Dublin City University

Development of A Hybrid Genetic Algorithm Based Decision Support System for Vehicle Routing and Scheduling in Supply Chain Logistics Management

BY:

SEYED MOHAMMAD SHAFI KHANIAN (B. Eng., M. Eng.)

SUPERVISIORS:

PROFESSOR, M.S.J. HASHMI,

DR. TAMAS SZECSI.

This thesis is submitted in accordance with the requirements of Dublin City University for the degree of Doctor of Philosophy.

Submitted to Dublin City University, March 2007

Declaration

Project Title:

Development of A Hybrid, Genetic Algorithm Based Decision Support System for Vehicle Routing and Scheduling in Supply Chain's Logistics Management.

Supervisors:

Professor M.S.J. Hashmi,
Dr. Tamas Szecsi.

I herby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy (PhD) is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged with the text of my work.

Signature: ______________________________

ID No: ______________________________

Date: 12/03/2007

Dedication

This thesis would be incomplete without a mention of the support given me by my parents to whom this thesis is dedicated. You kept my sprits up when though days failed me. Without you lifting me up when this thesis seemed endless, I wonder if at all this work would have ever been completed.

Acknowledgements

I would like to thank each one of the people who directly or indirectly contributed to the completion of this research.

Professor Hashmi, for the generous giving of his time, when I needed it most and also for his guidelines and patience over the past years.

Special thanks to Dr. Szecsi, who has provided me with patient directions throughout this work and his guidelines especially in the field of Genetic algorithms and its applications, which were absolutely critical in this work.

Also, I would like to thank the management board and logistic department within Fergas (Ltd.) liquefied petroleum Distribution Company in Iran in providing me necessary information on their supply chain activities.

To my lifetime friend Dr. John Ryan, who did not waste a minute to discourage me and radiate negative energy into me during the last couple of years. Thank you John for your great wisdom and knowledge.

To Dr. John Geraghty, who never failed to support me with my lack of understanding on manufacturing systems and industrial engineering concepts. Also thank you for all those corrections and directions. Above all thank you for being a good friend.

My special thanks and appreciations to my brother Shahab for all those valuable debates on operation research, simulation modelling and optimisation. I always remain a fan of your works. Thanks!

Also, to my younger brother Shakib, whom I shared a flat next to the college, thanks for all those great food that you never provided me. Thank you for putting up with me!

Special thanks to my soul mate, inspiration and friend Laila. Thank you for your patience, tolerance and acceptance. Thanks for all your encouragements and supports.

Thanks to the crowd in Intel Ireland, Philip Lyons, Paul Sheehy, Donna, Gemma, Mark and Nigel for all those free times you let me have to work on my thesis.

My special appreciations to my dearest friends, whom I have shared lots of good memories during my time in Ireland, including Dr. Ali Al-Aradi, Dr. Behrnam Heidari, Dr. Nabi Bakhtiari, Dr. Saleh, Dr. Hussain Askar, Dr. Amr Arisha, Dr. Tariq Mujber, Mr. Joe Bolger, Mr. Brian Denin, Mrs. Suzan, Mrs. Elain and Mr. Abdolhadee.

Abstract

By: S.M. Shafi Khanian

Vehicle Routing and Scheduling (VRS) constitute an important part of logistics management. Given the fact that the worldwide cost on physical distribution is evermore increasing, the global competition and the complex nature of logistics problems, one area, which determines the efficiency of all others, is the VRS activities. The application of Decision Support Systems (DSS) to assist logistics management with an efficient VRS could be of great benefit. Although the benefits of DSS in VRS are well documented, however in practice many organisations perform these activities manually using combination of skills, intuition and expertise.

A comprehensive review of literature revealed several drawbacks in the existing methods for addressing VRS. The traditional optimisation approaches have very limited applications and these require high computation time. Also, heuristic approaches are capable only to specific variation, a slight difference in the structure of the problem make the algorithm inefficient. Furthermore, metaheuristics methods require higher computation time and they are context dependent. Also, further investigations on the VRS problem formulations suggest that heuristic approaches usually address a single objective of distance minimisation. However in the real world there may be a number of conflicting objectives. In general, there is a lack of considerations for route selections, resource utilisation, unfulfilled demands, underused capacities, reliability of deliveries, fleet size, human fitness and operational cost. Also, these approaches fail to realise non-linearity within objectives and constraints defined for VRS problems. Furthermore, there are no clear distinctions between hard and soft constraints considered in these methods. Finally, the existing approaches fail to capture stochastic and dynamic nature of the logistics processes.

In order to overcome the above-mentioned drawbacks, this study designed and developed a hybrid DSS to assist logistics managers with VRS tasks. The capabilities of the developed DSS have then been applied to a Liquefied Petroleum Gas (LPG) distribution company. The architecture of this DSS is composed of Genetic Algorithm (GA) optimisation tool and a simulation model. The GA module aims to provide a pool of near optimum

transportation schedules. The simulation module is used to further evaluate the generated schedules. The feed back from the simulation module is used to update the GA for re-optimisation. Some unique features of this DSS are such as: development of a multi modal genetic algorithm to address VRS problems; considering supply chain performance measures as part of VRS problem formulation; allowing consideration of different objectives, soft or hard constraints concerning the supply chain, considering linear/non-linear relationships within objectives and constraints defined and finally, considering stochastic and dynamic behaviours of the supply chain system.

The GA and simulation tool integration provides unique benefits that have not been in the literature such as consideration of practical requirements, uncertainties, dynamic and stochastic behaviours, considering several criteria and producing different alternative solutions. Also, this integration allows the GA model to filter out solutions that are less competitive and therefore reducing the simulation time evaluation, which is computationally expensive. Furthermore, the human interaction with the system assists in generating higher quality of solutions. Finally, the clear benefit of this DSS is the fact that it greatly influences the applicability of the GA generated schedules and provides better confidence in implementation of these solutions.

Contents

List of Figures

List of Tables

List of Abbreviations

No.	Items used	Abbreviation
1	Agent Based Modelling and Simulation	ABMS
2	Air Pollution	ECA
3	Analytical Hierarchy Process	AHP
4	Approximation Method	VAM
5	Artificial Intelligence	AI
6	Asset Management Tool	AMT
7	Distributed Discrete Event Simulation	DDES
8	Dynamic Genetic Algorithms	DGA
9	Dynamic programming	DP
10	Enterprise Resource Planning	ERP
11	Environmental Cost	EC
12	Expert Systems	ES
13	First-In-First-Out strategy	FIFO
14	General Pickup and Delivery Problem	GPDP
15	Genetic Algorithm	GA
16	Granular Tabu Search	GTS
17	Graphical Information System	GIS
18	Holding Cost	HC
19	Human Factor Costs	HFC
20	IBM Supply Chain Analyser	SCA
21	Independent Identically Distributed	IID
22	Integer Programming	IP
23	Inventory cost	INVC
24	Lagranean Relaxation	LR
25	Linear Programming	LP
26	Liquefied Petroleum Gas	LPG
27	Maximum Number of Generations	Gn
28	Mean Time Between Failures	MTBF
29	Mean Time To Repair	MTTR
30	Mixed-Integer Programs	MIP
31	Multi Population Genetic Algorithms	MPGA
32	Neighbourhood search	NS
33	Neural Networks	NNs
34	New Population	NewPop
35	Noise Pollution Cost	ECN
36	Non-Polynomial	NP
37	Non-standard Genetic algorithms	NGA
38	Object Linking and Embedding	OLEII
39	Object Oriented Programming	OOP
40	Old Population	OldPop
41	Pareto Optimal Genetic Algorithm	POGA
42	Partially Mapped Crossover	PMX
43	Physical Processes	PS
44	Probability of Crossover	Pc
45	Probability of Mutation	Pm
46	Purchase Orders	PO
47	Request for Quote	RFQ
48	Restricted Tournament Selection	RTS
49	Safety Cost	SAFC
50	Schedule Completion Time	SCT
51	Service Level	SL
52	Shortage cost	SC
53	Simple Genetic algorithm	SGA
54	Simulated Annealing	SA
55	Small to Medium sized	SME
56	Steady State Genetic algorithms	SSGA
57	Surface Transportation Economic Analysis Model	STEM
58	Supply Chain Management	SCM
59	Tabu Search	TS
60	Travelling Salesman Problem	TSP
61	Unit Periods of Waiting	UPW
62	Vector Evaluated Genetic Algorithm	VEGA
63	Vehicle Routing and Scheduling activities	VRS
64	Vehicle Routing Problems	VRP
65	Vehicle Scheduling Problem	VSP
66	Vehicle Scheduling Problem with Length of Path Restrictions	VSPLPR
67	Vehicle Scheduling Problem with Multiple Depots	VSPMD
68	Vehicle Scheduling Problem with Multiple Vehicle Types	VSPMVT
69	Vehicle Routing and Scheduling with Backhaul Time Windov	VRSBTW
70	Vehicle Routing and Scheduling with Time Windows	VRSTW
71	Washington Department Of Transportation	WSDOT
72	Water Pollution Cost	ECW
73	World Wide Web	WWW

Chapter 1: Introduction

1.1 Introduction

Logistics management as a component of the supply chain involves managing the flow and storage of materials and information across the entire organisation with the aim to provide the best customer service in the shortest available time at the lowest cost. Some of the problems related to logistics management may be considered as: selection of distribution channels, determination of customer service level, location planning, inventory management, transportation means selection, fleet composition, vehicle routing and scheduling [1].

Transportation is one of the key activities of logistics management that may vary from firm to firm. Transportation plays an important role in both the physical supply and distribution phases of the logistics. Companies have realised that organising the supply of incoming parts and out going goods can account for 10 % of their costs [1]. Also, several studies have reported that transportation, as the largest element in logistics costs, typically accounts for 5 % of the product value [2]. Therefore, companies besides reengineering and scrubbing the waste from their assembly lines have noticed logistics seems worthy of rather closer attention.

One area that determines the efficiency of transportation management is the Vehicle Routing and Scheduling activities (VRS). The vehicle routing is concerned about the movement of vehicles and the vehicle scheduling gives explicit consideration to time for vehicle movements. In general, most real world VRS problems are perceived difficult to manage. The difficulties are mainly due to the commonly inherited characteristics of VRS activities. The VRS problems are typically subject to:

- ***Large Decision Space:*** Most VRS problems consist of a large number of decision variables and the possible options/strategies available to satisfy a problem are difficult to understand, evaluate and prioritise.
- ***Uncertainty:*** this is an inherent feature of a typical VRS environment; decisions must be made with uncertain and incomplete knowledge about the future situations (e.g.

supply acquisition, resource availability and demand levels). In addition, unexpected operational events within the execution environment (e.g. supply delays, breakdowns) need to be handled.

- ***Complexity:*** VRS problems consist of numerous components characterised by their dynamic behaviour and high interconnectivity. Understanding the dynamic behaviour of such problems and solving them is usually difficult.
- ***Highly Constrained:*** in a supply chain, the production and logistics processes are driven by a diverse and often contradictory set of constraints ranging from production objectives (e.g. efficiency, quality, profitability) to physical constraints (e.g. resource capabilities, utilisation requirements, operating preferences).

The shear complexity of VRS activities will be even more evident with increased competition and changing demands in the marketplace. Given the complexity of such problems and their abundance, the application of computer systems, particularly decision support systems to address VRS problems is expected to increase significantly. Considering the potential benefits of such applications the early expectation was that organisations would adopt these systems with enthusiasm. However, very few organisations adopted such systems. Also these organisations were mainly large in size as reported in [3]. In practice, organisations tend to rely on schedulers to manage their VRS activities using combination of skills, intuition and experience.

The failure in application of such systems within organizations indicated that the conventional approaches to VRSPs were not addressing the problems actually faced by the logistics managers. Based on the survey conducted in this research work it was obvious that the conventional approaches presented a number of shortfalls. These approaches

- Failed to represent real situations,
- Required high computation time,
- Limited in applications,
- Context dependent,
- Failed to consider multiple conflicting criteria,
- Failed to supply alternative solutions,

- There is a lack of considerations for route selections, resource utilisation, unfulfilled demands, underused capacities, reliability of deliveries, fleet size, human fitness and operational cost.
- Failed to realise non-linearity within objectives and constraints defined for VRS problems.
- There are no clear distinctions between hard and soft constraints considered in these methods.
- Failed to capture stochastic and dynamic nature of the logistics processes.

Therefore, handling the VRS operations in the supply networks call for new approaches to better understand logistics processes and also new methods and models to deal with logistics operations and activities in a more effective way.

1.2 Research Objectives

To address the shortfall of the existing approaches, this research presents the design and development of a hybrid genetic algorithm based decision support system to assist logistics management with VRS activities. Some of the novel aspects considered in designing the proposed DSS are:

- Providing a flexible means to formulate both single and multi-criteria VRS optimisation problems. Also, allowing for real world constraints considerations.
- Considering all relevant supply chain logistics performance measures in setting the objective function formulation.
- Providing flexibility in defining linear and non-linear relationships both in terms of system's objectives and also constraints.
- Facilitating a better approach to delivery times, route selections, resource utilisation, unfulfilled demands, underused capacities, environmental, safety, reliability of deliveries, fleet size, human factors, inventory analysis and operational costs in VRS problems which are traditionally neglected in addressing these problems.
- Providing a more flexible way to model stochastic, dynamic and complex relationships, which exist in logistics systems, which are also neglected in conventional approaches.
- Providing alternative solutions with high credibility and reliability for real system applications to support decision-making.

1.3 Significance of Decision support systems

The development of an efficient DSS for managing VRS activities would lead to a more widespread application of this concept resulting in significant savings for the logistics industry. The potential benefits provided by such systems could be described as:

- ***Less dependence on personnel experience:*** this is one of the main benefits of such systems. The dispatching operations particularly in Small to Medium sized (SMEs) companies are manually managed and it is highly dependent on the experience of the manager in charge. This would cause major problems if the experienced personnel decide to leave this job either temporarily or permanently. Training of new staff tends to follow a long learning curve, which may be substantially shortened by the proposed DSS. Therefore, the aim of this DSS is to generate acceptable solutions, which serve as examples and will provide the opportunity for the investigation of alternative scenarios.
- ***Enhancing work environments:*** the proposed DSS considers a large amount of data look-up, computations, and constraints to provide alternative solutions. In this way it would provide a better work environment and faster response time.
- ***Improving decision quality:*** the DSS provides a means for testing and evaluation alternative solutions therefore it would improve decision quality. Finally, the aim is to build a DSS suitable for staff with minimum analytical skills to prevent any further dependency on data processing or operation research skills to keep the system running. In addition, the generated solutions could be used to assist decision making at different management levels:
  - **Strategic level**: the schedules provide the manager with transportation planning and fleet composition.
  - **Tactical level**: the schedules are used to indicate the allocation of resources to the general transportation plan.
  - **Operational level:** this level refers to the detailed scheduling of routes and tours on a daily basis.

1.4 Motivation

To examine the capability of the proposed DSS, it was decided to apply this tool to a real case distribution problem. For this reason Fergas (Ltd.) which is a Liquefied Petroleum Gas (LPG) Distribution Company was considered. This company represents a Small to Medium sized Enterprise (SME), involved with distribution activities. The main objective of this company is to provide customers with bottled LPG cylinders. For this reason, the company needs to supply its bottling plants with LPG from available refineries, which are dispersed geographically. The transportation is mainly carried out by means of a given fleet and the objective is to provide the required demand at the right time and the lowest possible cost. This company is located in Iran and it consists of a number of bottling plants to cover the need for LPG at different local markets. Figure 1-1, describes the LPG supply chain for this company. The following sections briefly describe the main components of this organisation.

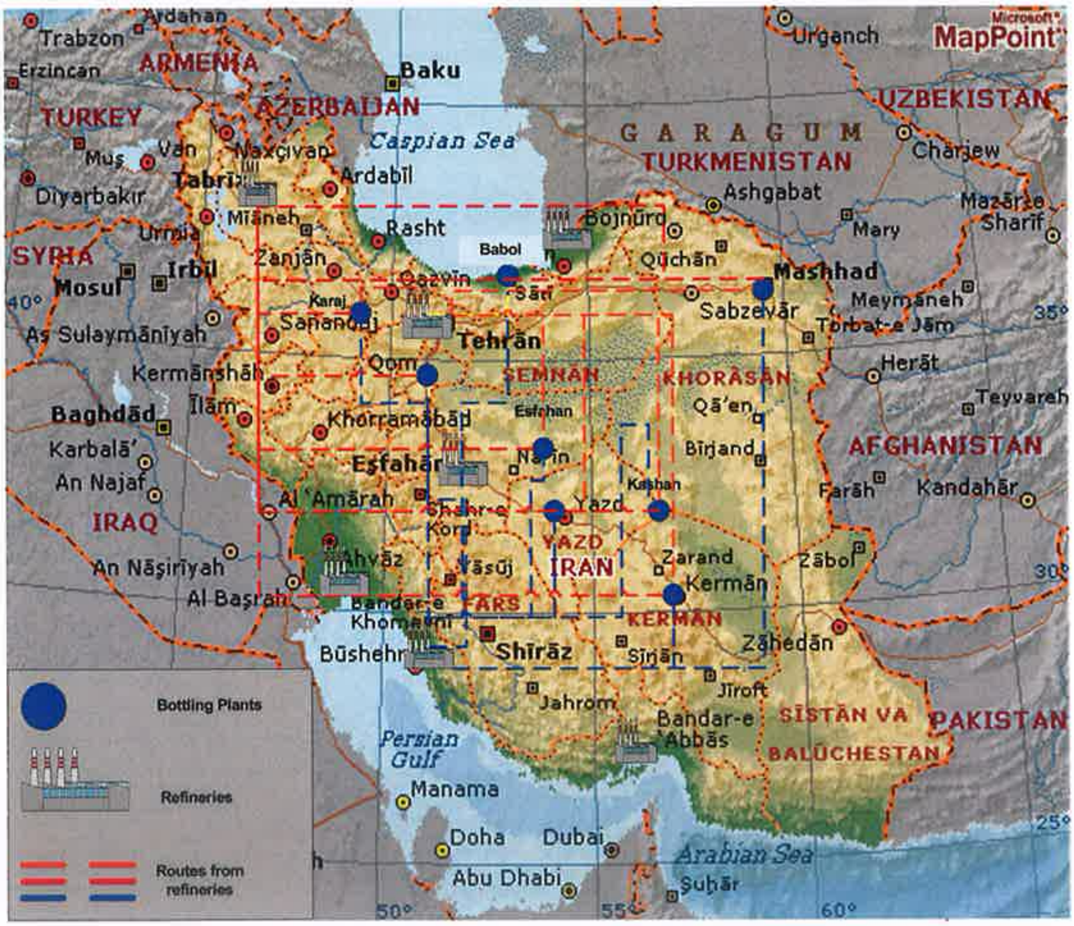


Figure 1-1 An overview of LPG supply chain

1.4.1 LPG Supply Chain

Figure 1-2 illustrates, the integrated supply chain for LPG within Fergas organisation. The main elements of this supply chain could be described as follows:

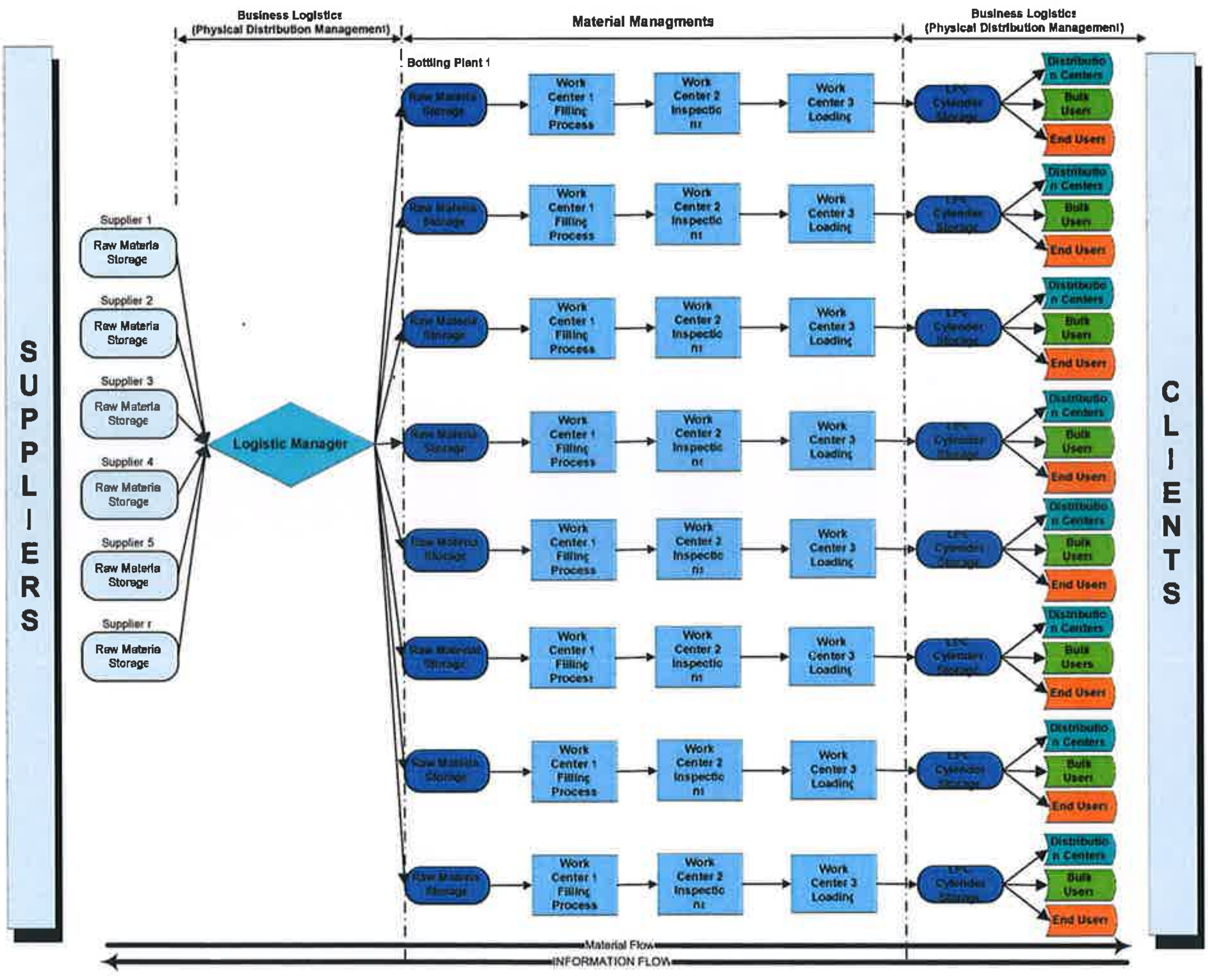


Figure 1-2 An Integrated Supply chain for LPG

1.4.2 Suppliers:

Suppliers are the gas refineries that are used to LPG to be used by different LPG processing companies. These refiners are located in different geographical locations and they produce variable and limited amount of LPG. There are a number of refineries that are used to provide LPG from but some of the main ones are located in cities known as Abadan, Bandar Abas, Esfahan and Tehran.

1.4.3 Bottling Plants:

The bottling plants are the LPG processing facilities. These facilities are used to process LPG into gas cylinders to be used by the end customers. These facilities are dispersed geographically. They are in charge of providing the demand for LPG in their respective local markets. In the considering supply chain there eight main bottling plants located in cities known as: Babol, Esfahan, Karaj, Kashan, Kerman, Qom and Yazd. The LPG consumption by each plant varies and it is mainly seasonal. As indicated in Figure 1-3, the demand for LPG is usually high during the cold seasons (winter & fall) and low during the warm seasons (spring & summer).

Each facility is equipped with LPG storage reservoirs, filling equipments and material handling equipments. As shown in
Figure 1-2, the LPG is withdrawn from the storage facilities as needed by the production centre, which in this illustration comprises three work centres as follows:

- **Work centre 1:** Here the raw material, which is LPG, is compressed and it is inserted into 11 Kg gas Cylinders. The inputs to these stations are LPG and Gas Cylinders. The output is the finished products, which are sent to the next operation.
- **Work Centre 2:** Here the Gas cylinders are inspected for any defects and Leakages. Upon the acceptance, the cylinders are sent to the next stage.
- **Work Centre 3:** The accepted cylinders are the finished products, which are stored in the finished product storage.

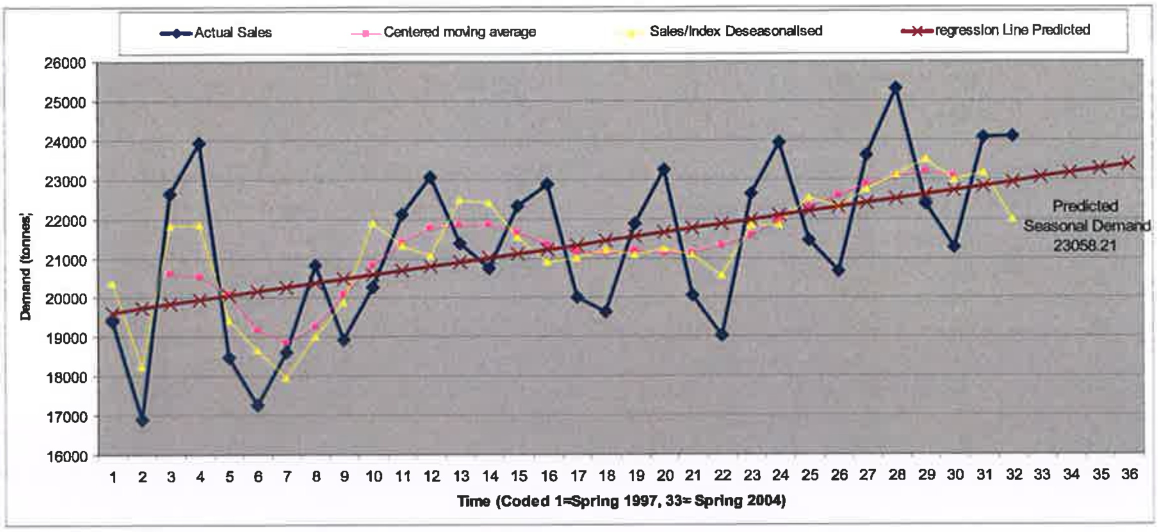


Figure 1-3 Forcasted seasonal demands for LPG

The deliveries are accomplished with surface transportation such as trucks. These trucks could have different storage capacities. Trucks are typically hired and there are also some trucks owned by the company for transportation.

Introductions

One of the main tasks of the logistics manager is to provide each processing facility with their required demands of LPG. The manger is faced with a dynamic environment, where supply sources, processing plants and resources are subject to stochastic behaviours. Based on the demands, the manger determines the routes and schedules on a daily basis for a number of vehicles from refineries, to supply processing plants to maintain an acceptable service level and at an acceptable cost to the company.

There exist a number of operational constraints in the real system that impacts the effective resource planning. Some of these factors are manpower availability, vehicle availably, environmental restrictions such as weather and road conditions, traffic, human factor restrictions, supply availability, processing plant breakdowns, inventory levels and delivery time restrictions.

In managing the transportation schedule, there are several criteria or objectives for assessing a transportation schedule. In this approach, typical logistics performance measures are considered. Some of these parameters considered are:

- **Fleet size:** The number of vehicles used in a transportation schedule strongly affects the strategic investment planning, due to the high cost of trucks involved in the transportation. Therefore, reducing the number of required trucks could be useful in reducing logistics costs.
- **Travelling distances:** Minimising the mileage could directly reduce the operational cost related to fuel and drivers' time. Also minimising the number and length of deadheading trips could directly impact the operational cost.
- **Inventory Levels:** Considering the target inventory levels at each processing plant, the transportation schedule should minimise the deviation from these target values to prevent any additional holding or shortage costs.
- **Resource Utilisation:** The efficient routing and scheduling of resources may well increase the maximum operating time of the resource by eliminating idle time from the system.
- **Service level:** Measures such as meeting the demand level of the processing plants and delivery times must be considered to improve customer satisfaction.

- **Environmental Impact:** To measure the impact of transportation schedules on the environment, three main aspects are considered air, water and noise pollution. The environmental impact is measured based on the routes taken and speed of travel. The aim is to minimise this negative impact by taking more environmentally friendly trips.
- **Completion time:** The schedule completion time reflects the operational costs, labour costs and loss of customer good will. The aim is to minimise completion time to handle all the transportation within the specified time horizon.
- **Safety:** Typically the cost of vehicle accidents can be quite high, particularly in the case of butane gas, which is highly explosive. This cost includes vehicle and other property damage as well as personal injury and mortality. The aim is to minimise safety cost by assigning trucks to routes that are more favourable to safety measures.
- **Human Factors:** In assigning routes and schedules to personnel for transporting propane gas, one of the primary objectives is the convenience of the crew. For this reason the aim is to perform the transportation in a way to minimise the total inconvenience of the crew.

1.5 Problem Definition

The cited problem represents logistics management for the upper stream of the considered supply chain, which involves dispatching raw material from supply sources to the processing plants based on monthly demands. This problem can be considered as a variation of VRSPs, which are known to be Non-Polynomial (NP) [4, 5] combinatorial optimisation problems. In general, it is very difficult to solve these combinatorial problems to optimality and their computational burden exponentially grows with the problem size. This makes it clear that the dispatching decisions are too complex for manual optimisation [6, 4].

There are a number of conflicting objectives in this problem. On one hand there is a need of minimising factors such as transportation and inventory costs and on the other hand there is a need for maximising resource utilisation, performance and customer service. Using the proposed DSS, the LPG dispatching problem could be formulated in its most general form as a Multi-objective, non-linear optimisation problem. The aim is to provide the logistics manager with a decision support tool to generate a set of different optimum transportation schedules. In this way the decision maker can evaluate some alternative solutions before finalising on a dispatching policy, therefore, providing a more flexible decision support system.

1.6 Approach of Thesis

To meet the initial objectives of this research work, a comprehensive survey was conducted on the current state of art in VRS problems. To this effect different categories of VRSP were recognised, different approaches and VRS problem formulations were studied. Based on the survey conducted a hybrid optimisation based decision support was designed and developed to assist management with decisions on VRS activities. This hybrid approach is composed of both a search engine and simulation module. The search engine was developed based on a genetic algorithmsearch. The aim of this module was to provide alternative optimum solutions. The simulation module was used to evaluate different alternative solutions. The feed back from the simulation module was used to update the GA for re-optimisation.

This research was further extended to examine a typical organisation involved with VRS activities. For this purpose the supply chain and manufacturing facilities of a LPG distribution company were investigated. Using this proposed DSS the VRS activities within this organisation were formulated in their most general form as a multi-objective non-linear optimisation problem.

1.7 Outline of Thesis

This chapter briefly described the shortfalls of the existing approaches in addressing VRS activities. It then proposed the design and development of a hybrid decision support system to assist VRS activities. Moreover, a brief description of the considering supply chain used for this application was provided. The remainder of this chapter outlines the main topics discussed and developed in the proceeding chapters.

- ***Chapter 2 Literature Survey:*** This chapter provides a comprehensive literature review on the concept of VRSP and it defines different categories of such problems. It describes optimisation techniques, multi-objective optimisation, applications of heuristics and Meta heuristics in VRSP and finally, Artificial Intelligence and Simulation modelling are introduced as alternative approaches to handle VRSPs.

- ***Chapter 3 Methodology:*** This chapter proposes an architectural design for a DSS to address the considered logistics problem. Different components of this DSS are described. In particular, the concepts and principles of the genetic algorithm are explained. In addition, an investigation is conducted to see how such GAs are used to formulate VRSPs. To this effect the GA application in fields such as transportation, delivery, vehicle scheduling, vehicle routing and logistics problems were explored. Based on this survey the shortcomings of these applications are discussed in respect to the considered problem. Also, a Pareto Optimal GA used to address multiple criteria problems is introduced and a hypothetical example to address this problem is presented.
- ***Chapter 4 Search Engine Design and Development:*** This chapter describes how the search engine was developed. It describes the random search method implemented; the genetic algorithm approach and Pareto based genetic search method. Finally, it demonstrates how the different components of GAs were developed. In addition the main objectives in this system are identified and problem formulation using these criteria is presented. Furthermore, the considered system constraints and the penalty methods used to allow violation of constraints are presented in this chapter.
- ***Chapter 5 Simulation Modelling:*** This chapter provides some background on the application of simulation modelling in manufacturing. It particularly looks at the application of simulation modelling in logistics and supply chain management. It further classifies the simulation tools available and it introduces the basic concept of the Witness simulation-modelling tool, which was used to model the considered system. Finally some recent advances in simulation modelling techniques are introduced.
- ***Chapter 6 Simulation Model Development Using Witness*:** This chapter first describes the supply chain system under study. It identifies different components of this system and describes how a simulation model was developed for this system. Also, it discusses how the supply chain model was verified and validated.

- ***Chapter 7 Results:*** this chapter used to provide a detailed account of the series of experiments carried out on the GA search engine. It describes how the initial parameters for this GA application were set. It identifies the impact of population size, selection methods, crossover and mutation operators and Elitism on the GA performance. It further illustrates how the simulation model was used to further analyse the generated schedules and finally it highlights the feed back from the simulation module to the GA search for the re-optimisation. The results of such experimentations are documented in this chapter.
- ***Chapter 8 Conclusions and further works:*** This chapter provides a conclusion on the achievements of this work and it further identifies how this research can be extended.

Chapter 2: Literature Survey

2.1 Introduction

Transportation is a key activity of logistics that may vary from firm to firm. Logistics may be defined as "the provision of goods and services from a supply point to a demand point" [7]. Transportation plays an important role in both the physical supply and distribution phases of logistics. According to some estimates the European logistics market was valued at about $155 billion in 1999 and it is expected to expand to $213 billion by 2005 [8]. Therefore, companies realised logistics as a major filed of improvements and to gain competitiveness in the challenging global market.

In addition, logistics networks are becoming evermore complex mainly due to the greater demands from customers and increased competition from competitors. These elements have left pressure on companies to focus on delivering greater value to the customers, in less time. One area that determines the efficiency of logistics management is the Vehicle Routing and Scheduling activities (VRS). The problems that managers confront in their efforts to manage VRS processes in supply networks are mostly perceived as uncertain, non-linear, and increasingly complex. Therefore handling the VRS operations in the supply network would call for new approaches to better understand logistics processes and also new methods and models to deal with logistics operations and activities in a more effective way.

To this effect, the aim of this chapter is to provide a general review on the studies carried out in the earlier chapter on the current approaches and possible new approaches to support better understanding and managing of VRS operations.

As VRS problems are highly regarded as NP-Hard optimisation problems, this chapter starts with an overview of the concept of optimisation. It further, studies the application of heuristics as possible methods to produce optimum solutions. It explains about different strategies that could be used in developing heuristics and it identifies a number of useful methods for this purpose. This chapter classifies different types of vehicle routing and scheduling problems. Moreover, it identifies different possible problems within each category. Also, it expands on the classical and metaheuristic approaches to address these problems. It finally describes the possible enabling techniques and approaches that could be further used to understand and address VRS activities.

2.2 Optimisation Problems

The optimisation problems are generally classified into two categories [9] those with continuous variables, and those with discrete variables, which are known as combinatorial. The continuous problems are generally concerned with a set of real numbers or even a function.

On the other hand, the combinatorial problems are looking for an object from a finite set of numbers such as integer, permutation or graph. In general, the combinatorial optimisation problems are concerned with the efficient allocation of available resources to meet desired objectives when the values of some or all of the variables are restricted to be integral [10]. Also, there are usually constraints on resources such as labour, supplies or operations that confine possible feasible alternatives. The aim in such problems is to find the best possible alternative solution.

The combinatorial optimisation is defined as the process of finding one or more, best (optimal) solutions in a well defined discrete problem space [10]. There are many problems in management and engineering disciplines, which can be formulated as combinatorial optimisation problems as they can have only a finite number of alternatives feasible solutions. This reflects the versatility of the combinatorial problems.

Also, optimisation problems can be categorised as either linear or non-linear optimisation problem. Linear optimisation problems deal with a linear objective function to be maximised or minimised subject to linear restrictions or constraints on the non-negative variables. Any nonnegative solutions of the constraints are known as feasible solutions. Any feasible solution maximising or minimising the objective function is called optimal. However, in many real world applications there are no linear relationships within the parameters. Here in the considered problem the transportation cost can be either linear or non-linear. If the transportation cost is directly proportional to the travelled distance then the objective would be linear. However there are cases where transportation cost is not proportional to the travelled distance due to different rating schemes. In this way the objective function or the constraints are non-linear and therefore it is called a non-linear optimisation problem.

Furthermore, an optimisation problem may be composed of a single objective or a number of objectives. In single objective problems, the goal of a solution is the identification of the optimal solution, which is the feasible solution that gives the best value of the objective function. However, the concept of optimality is not valid, when considering multi criteria (objective) problems. Because a solution, which maximises one objective, will not in general, maximise any of the other objectives. In multi-objective optimisation, the optimal solution is referred to as a non-inferior solution. Traditionally, there are three approaches in addressing multi-objective problems that are described briefly in the following sections.

2.2.1 Weighting-Based approach:

Weighting the objectives to obtain non-inferior solutions is the oldest multi-objective solution technique. Zadeh [11] was the first to recommend the use of weights to approximate the non-inferior set. Considering an objective function for a multi-objective problem as:

$$\text{Maximise } Z = [Z_1, Z_2]$$

If there is a way to assign the judgment value that one objective costs ***w*** Euros, then the multi-objective problem could be reduced to a single-objective problem. The specification of ***w***, known as weight on objective Z_2, is similar to the identification of a desirable trade-off between Z_1 and Z_2. Based on this concept, the objective function can be formulated as:

$$\text{Maximise } Z(w) = Z_1 + wZ_2$$

Therefore, the new objective function has single dimension and is denoted by Z [*w*] to signify the dependence of the new function on the value of the weight ***w***. The weight reflects a decision maker's preference with respect to the importance of the considered objectives. Given a single objective function and a set of constraints, and using a linear programming approach an optimal solution can be generated.

2.2.2 Constraints based approach:

In this approach, the objectives are implemented as constraints, and feasible solutions are searched relaxing progressively the constraints that represent the objectives with lower priority. Given a multi-objective problem with ***P*** objectives (i.e. $Z_1,...,Z_p$), the mathematical formulation would be:

$$\begin{aligned} & Maximise Z(x_1, x_2,..., x_n) \\ & = [Z(x_1, x_2,..., x_n), Z_2(x_1, x_2,..., x_n),..., Z_p(x_1, x_2,..., x_n)] \\ & s.t.(x_1, x_2,..., x_n) \in F_d \end{aligned}$$

The constraint formulation of the above multi-objective problem is

$$\begin{aligned} & Maximise Z_h(x_1, x_2,..., x_n) \\ & s.t.(x_1, x_2,..., x_n) \in F_d \\ & Z_k(x_1, x_2,..., x_n) \geq L_k \\ & k = 1,2,..., h-1, h+1,..., p \end{aligned}$$

Where the h^{th} objective was arbitrarily chosen for maximisation. This formulation is a single-objective problem, so it can be solved by conventional methods in linear programming.

2.2.3 The goal programming approach:

In the goal programming approach, the aim is to minimize one objective while constraining the remaining objectives to be less than given target values. This method is especially useful if the user can afford to solve just one optimisation problem. However, it is not always easy to choose appropriate "goals" for the constraints. The goal programming cannot be used to generate the Pareto set effectively, particularly if the number of objectives is greater than two [16].

2.2.4 Pareto Optimality:

A notation of optimality that respects the integrity of each of the separate criteria is the concept of Pareto Optimality [12]. A feasible solution to a multi-objective programming problem is ***Pareto optimal*** if there exists, no feasible vector of decision variables which would improve some criterion without causing degradation in at least one other criterion [11]. This concept does not provide a single solution, but rather a set of solutions called Pareto ***optimal set***. The feasible solutions included in the Pareto optimal set are called ***non-***

dominated. The plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is called the ***Pareto front***.

The Pareto optimal set also known as the ***non-inferior*** set generally includes many alternatives. It is not possible to consider all the alternative solutions as the final optimal solution. In fact, the decision makers prefer a solution amongst the available alternatives. This preferred alternative is called the ***best-compromise solution***. The purpose of the Pareto optimality is to find the optimal frontier, which dominates the other solutions. These solutions are not worse with respect to every object, but rather better for at least one objective.

2.3 Optimisation Techniques

Some of the basic solution techniques for combinatorial optimisation problems are such as integer programming, dynamic programming, and heuristic problem solving. These approaches are outlined briefly below.

2.3.1 Integer Programming:

Combinatorial problems are often referred to as integer programming models. When, formulating *Linear Programming* (LP), there are usually certain variables that could take integer values but they are left to take fractional values. *Integer Programming* (IP) is a numeric solution in which the variables take integer values. IP occurs frequently due to the fact that many decisions are essentially discrete in that one or more options must be chosen from a finite set of alternatives. Also, the problems, in which some variables can be integer and some variables, can be fractional are addressed by Mixed-Integer Programs (MIP) [10].

There are at least three different approaches for solving integer-programming problems; these are typically combined into hybrid solution procedures in computational practice. These techniques are namely known as Enumerative, Relaxation and decomposition and cutting plane approaches [10]. These techniques are further presented in Appendix A.

2.3.2 Dynamic Programming (DP):

Another method of solving optimisation problem is dynamic programming (DP). It requires being able to express or compute the optimal solution to a given problem instance $\boldsymbol{I}$ in terms of optimal solutions to smaller instances of the same problem. This is called problem decomposition. The optimal solutions to all the relevant smaller problem instances are then computed and stored in tabular form. The smallest instances are solved first, and at the end of the algorithm, the optimal solution to the original instance *I* is obtained. This method can be considered as a bottom-up design strategy [14].

2.3.3 Heuristics:

The IP and DP approaches are designed to produce global solutions for the problems, which they are applied. However, many real world problems are so large and difficult that these methods cannot achieve this solution effectively due to their large storage or computational time requirements. In this case a more moderate approach is required, where instead of obtaining an absolute optimum, it is often more realistic to design a solution procedure which will produce, in reasonable computing time, solutions which are relatively close to the optimum. Such procedures are known as heuristics [15].

Heuristics are methods of performing minor modifications, or a sequence of modifications to a given solution or a partial solution. The actual modifications that are done will involve a neighbourhood search. A neighbourhood of an element X, are those elements, which are similar to X by some means. In general a heuristic algorithm will consist of iteratively applying one or more searches, in accordance with a certain design strategies.

2.3.4 Design Strategies for heuristic algorithms

The design strategies involve designing a neighbourhood search and incorporating it into a heuristic search algorithm. This is a widely used method in solving combinatorial optimisation problems. A fundamental idea of heuristic methodologies is that of neighbourhood search (NS) [15]. In the context of this search method, a solution is specified by a vector x, where the set of all (feasible) solutions is denoted by X. The cost of solution x or the objective function is denoted by $C(x)$. Each solution $x \in X$ has an associated set of neighbours $N(x) \subset X$, called the neighbourhood of x. Each solution $x' \in N(x)$ can be reached directly from x by an operation called a *move*. X is said to move to x', when such an operation is performed. Here, the choice criteria for *selecting moves* and *termination criteria* for ending the search are specified externally. Specifying these parameters in different ways would alter the method to yield a variety of procedures. The main shortfall of this method is its tendency to deliver solutions, which are only local optima [15].

One must notice that there is a trade off that must be considered in any heuristic algorithm. If a large neighbourhood is considered then, it is expected that the given search space contains a better solution than using a small neighbourhood. However, this would reflect the penalty to be paid in terms of computation time, if the neighbourhoods are too large. A large amount of work has been carried out on heuristic methods for solving combinatorial problems. There are four basic strategies for heuristic procedures. Many methods comprise a combination of more than one of these strategies [16]. These methods are described in Appendix A.

2.3.5 General purpose Heuristics

Heuristics can be categorised into problem specific and general-purpose heuristics. As evident, the problem specific algorithms are tailored to one particular problem, which would not work for different ones. However, the general heuristics can be easily tailored to solve any combinatorial optimisation problem. The following sections present a number of these heuristics. Also, the Gradient, Random and Iterated Local Search methods are covered in Appendix A.

2.3.5.1 Simulated Annealing (SA)

The ideas that form the basis of SA were first published by Metropolis et al. [17] in 1953. Thirty years later, Kirkpatrick et al. [15] suggested that this type of simulation could be ued to search the feasible solution of an optimisation problem, with the objective of converging to an optimal solution. SA is a neighbourhood search technique that has produced good results for combinatorial problems. This algorithm begins by generating an initial solution (S) at random. At each stage the new solution (S') taken from the neighbourhood of the current solution is accepted as the new current solution, if it has a lower or equal cost. Otherwise, the new solution is accepted with a probability $e^{-\Delta/T}$, where Δ is the difference between the costs of s and s', and $T \in R$ is referred to as temperature.

Initially, T takes a user-defined value, and iteration by iteration is decreased according to some temperature scheme referred to as a cooling schedule. Thus at start of SA, many inferior moves are accepted, but at the end only improving ones are likely to be accepted. This method converges to a local optimum as the temperature approaches zero, but because SA has performed many iteration at a high temperature, this may have pushed the search path into new areas, where a better local optimum solution could hopefully be reached. The algorithm terminates when a suitable stopping condition is satisfied (usually in some zero or near-zero value of the temperature parameter). The generic decisions to be considered in implementing the SA in practice include: Initial Temperature, Cooling Schedule and Final Temperature. Also, Nearchou [18] suggests that after from the temperature and cooling schedule, the performance of this method could be influenced by factors such as: *stopping condition, the choice of feasible solutions, the form of the objective (cost) function, the way of choosing a neighbour structure.*

2.3.5.2 Tabu Search (TS)

Tabu search is a local search Metaheuristic proposed independently by Glover et al. [19]. This method is like Simulated Annealing (SA), which is based on neighbourhood search that aims to prevent convergence to local optima in rather a deterministic way, which tries to model the human memory processes.

This method explores the solution space by moving at each iteration form a solution *(s)* to the best solution in a subset of its neighbourhood *N(s)*. In this way, the current solution may deteriorate from one iteration to the next. Therefore, to avoid cycling, solutions possessing some attributes of recently explored solutions are temporarily declared Tabu or forbidden, unless their cost is less than a so-called aspiration level. Some implementations allow for intermediate infeasible solutions. Also, various techniques are often employed to diversify the search process. These are [20]:

- **Recency:** Recency-based tabu search encourages exploration of parts of the solution that have not been visited previously. This is achieved by prohibiting the reversal of most recent moves. Recency simulates short-term memory.
- **Frequency:** using frequency measures such as: residence and transition measures would generate penalties, which modify the objective function of the search. This

measure would encourage diversification by the generation of solutions that represent a combination of attributes significantly different from those previously observed. Frequency provides a long term memory for the search mechanism.

- **Quality:** this is used to refer to those solutions with good objective function values. Providing of such solutions may be used to stimulate a more intensive search in the general area of these elite solutions.
- **Influence:** this is a measure of the degree of change made in a solution structure. This is used in aspiration criteria and also in the development of candidate list strategies.

2.3.5.3 Genetic Algorithms (GA):

The GA as search procedures is based on mechanics of natural selection and natural genetics. In its most general form, a GA creates a population of feasible solutions and generates future populations based on natural selection. Usually, the simulated annealing or tabu search algorithm begin with an initial feasible solution and proceed to construct from it a sequence of feasible solutions by applying a heuristic, which is in turn based on a neighbourhood search technique. In GA, the algorithmstarts with an initial population of feasible solutions, then feasible solutions from this population are recombined to produce offspring. After the children are obtained, a mutation operation is allowed to occur. This procedure produces the next generation of the population. The process can be iterated through as many generations as desired.

2.4 Classification of Routing and Scheduling Problems

The managerial decisions concerning the configuration of vehicle movements are classified as Vehicle Routing Problems (VRP) in the literature. These problems usually involve the specification of a sequence of locations that a vehicle must visit. When explicit consideration is given to the times at which various locations are to be visited, the problem is considered as a Vehicle Scheduling Problem (VSP). In many cases the routing and scheduling problems interact and result in combined Vehicle Routing and scheduling Problems" (VRSP) [6].

2.5 Complexity of Routing and Scheduling Problems

Alter [21] suggests that a system's complexity is a function of the number of differentiated components that exist in the system and the number and nature of their interactions. In view of this definition, it is evident that the VRS is a complex process for distribution system considered in this study. In fact, this complexity is mainly due to the existence of many parameters and constraints concerning manpower, vehicles, environments and the other constraints that exist in this system.

As mentioned earlier most of the real life VRSPs are computationally complex and belong to the NP-Hard problems. It is very difficult to solve these combinatorial problems to optimality and their computational burden exponentially grows with increase in the problem size [6]. As a result researchers [22] have concluded that VRSPs present a prime candidate for analysis by heuristics. Heuristic algorithms offer a number of advantages in handling VRSPs [23]:

- They are able to handle efficiently a large number of constraints and parameters that exist in a routing and scheduling problem.
- They perform a relatively limited exploration of the search space and generally produce good quality solutions within modest computing time.

The following sections further describe different categories of vehicle routing and scheduling problems and heuristics for these problems are presented in section 2.9.

2.6 Vehicle Routing Problems (VRP)

The Vehicle Routing Problem (VRP) can be described as the problem of designing optimal delivery routes from one or several depots to a number of geographically scattered demand centres, subject to side constraints. The VRP plays a central role in the fields of physical distribution and logistics. There exists wide variety of VRPs and a broad case of literature on these problems. These problems can be grouped into the following categories [6]:

2.6.1 Separate and Single Origin and destination points:

In these problems a network is represented by links and nodes, where the nodes are connecting points between links, and the links are the time or distances to traverse between nodes. Typically, these problems involve a well-defined sub-set **P** of paths between a given pair of nodes. A subset **Q** of **P** has to be determined once paths in **Q** are preferred to any other path of **P/Q**, when some given criteria are considered. These problems are generally studied under the general title **Optimal Path Problems,** which is a very large grouping of network optimisation problems. Some of the basic categories describing this class of VRPs are such as the ***Shortest Path Problem***, ***Shortest-Path Models with Fixed Charges***, ***K-Shortest Path,*** and ***Minimal Spanning Trees***. These categories are further detailed in Appendix A. Usually once the number of paths in P is very large finding the shortest path requires high computation time and therefore methods to solve the problem in reasonable time are required. Some approaches to these problems are suggested in section 2.9.1.

2.6.2 Multiple Origin and destination points:

When there are multiple source points that may serve multiple destination points, there is a problem of assigning destinations to sources as well as finding the best routes between them. This problem commonly occurs when there is more than one supplier to serve more than one demand centre for the same product. This is further complicated when the source points are restricted in the amount of the total customer demand that can be supplied from each location. This type of problem is frequently solved by applying a special class of linear programming known as the *transportation method*. Some of the heuristic approaches to this problem include the Simplex Method, the Northwest Corner Method, the Vogel Approximation Method (VAM) and the Hungarian Algorithm [24].

2.6.3 Coincident Origin and destination points:

The logistic management frequently encounters routing problems in which the origin point is the same as the destination point. This class of routing problem commonly occurs when transport vehicles are privately owned. Some basic heuristics to these problems are presented in section 2.9.1. Some of the basic categories of this type of VRP are such as the ***Travelling Salesman Problem (TSP)***, ***Chinese Postman Problem***, ***Single Depot Multiple Vehicle Routing*** and ***Multiple Depot Multiple Vehicle Routing Problems***. These are further described in Appendix A.

2.7 Vehicle Scheduling Problems (VRSP)

Vehicle scheduling problems can be considered as routing problems with additional constraints having to do with the times when various activities may be carried out. The routing problem described in the last section places special importance to the spatial characteristics of activities performed. In scheduling problems, however, a time is associated with each activity. Therefore the temporal aspects of the vehicle must be considered explicitly. The feasibility of an activity is influenced by both space and time characteristics. Heuristics for these problems are presented in section 2.9.3. Bodin et al. [6] classified vehicle scheduling based on real world constraints. These constraints determine the complexity of the vehicle scheduling problems and can be specified as:

1. A constraint on the length of time that a vehicle may be in service before it must return to the depot for servicing or refuelling.
2. The restriction that certain tasks can only be serviced by certain vehicle types,
3. The presence of a variety of depots where vehicles may be housed.

Based on these constraints the VSP are categorised as follows and further explanation on each category is provided in Appendix A [6]:

- **The Single Depot Vehicle Scheduling Problem (VSP),**
- **Vehicle Scheduling Problem with Length of Path Restrictions (VSPLPR),**
- **Vehicle Scheduling Problem with Multiple Vehicle Types (VSPMVT),**
- **Vehicle Scheduling Problem with Multiple Depots (VSPMD),**

2.8 Vehicle Routing and Scheduling Problems (VRSP)

As Bodin et al. [6] described most of the combined routing and scheduling problems are characterised by task precedence and time window constraints. Task precedence relationships force the pickup activity for a task to precede the delivery activity for the task and the pickup and delivery tasks must be on the same vehicle.

A second set of constraints involves the servicing of tasks within specified time windows. A time window on a service task requires that the task be serviced within the specified time interval. For instance, considering a particular delivery that must be between 8:00 to 9:00

AM. Therefore any route, which involves this particular task, must ensure that the delivery time falls within these time bounds.

Considering no time windows, the set of tasks that may follow a particular task can be specified as a *priori* and one can construct a network including all the tasks. However, with time windows, the complete set of tasks that can feasibly follow a given task cannot be specified beforehand since the exact time of service for a given task cannot be specified in advance. Therefore, with time windows it becomes very difficult to construct the complete network of possible connections. A very well known problem involving both routing and scheduling is dial-a-ride problem as described in Appendix A for further readings.

2.9 Approaches to VRSPs

Over the past 50 years considerable research effort has been put into developing heuristics for solving specific variations of the VRSP. The available heuristics can be classified into two main groups, namely classical heuristics and metaheuristics as described in the following sections.

2.9.1 VRSP Classical Heuristics:

In the 1950s, early works [13, 25] concentrated on formulating VRSPs and how to obtain optimal solutions for these problems. Many routing and scheduling problems were formulated as a special class of Zero-One integer programming known as set partitioning or set covering problems. However, the lack of both efficient solution procedures and computing power prevented the tackling of real problems [3]. By the 1960s zero-one programming formulation could be solved in principle. However, the need for excessive calculations for small problems limited any potential applications. [26] concluded that this formulation was too complex to be useful for even non-trivial size VRSPs. Also [27] provided three exact algorithms to address VRPs. These algorithms used the branch and bound method based on bounds derived from the shortest spanning tree with fixed degree at one vertex and the minimal q-routes. Although, these algorithms were able to find exact solutions, it must be noted that this was mainly due to the fact that the problems were of small size and relatively free of constraints.

Heuristic algorithms for the VRSPs can often be derived from procedures derived from the TSP [28]. However, when applying these procedures care must be taken to ensure that only feasible solution is created. Some of the approaches developed to address VRSP are as follows:

- The Clark and Wright algorithm: This classical algorithm was first introduced in 1964 to solve capacitated VRPs in which the number of vehicles was unlimited. The method starts with vehicle routes containing the depot and one other vertex. At each step, two routes are merged according to the largest saving that can be generated.
- In 1965 [29] as part of addressing the TSP introduced a local search algorithm that was based on the notation of k-exchange. A number of researchers [30, 31] applied this approach to the basic variations of VRSPs and reported on its efficiency for solving related problems.
- In 1969 [32] developed the 3-Optimal method of route planning. This method was based on the principle that the established route configuration is optimal and any replacement of the links will generate a sub optimal situation.
- In the 1970s [33] solved the VRSP incorporating the time window aspect of the problem. In 1976, the Generalised Saving Method [34] and the Heuristic Tree Search Algorithm [35] as OR techniques achieved marginal improvements over the Saving and 3-Optimal methods.
- The sweep algorithm for VRP [36]: This method is commonly attributed to Gillett and Miller (1974). This method represents vertices by their polar coordinates. It established individual routes for each vehicle, by assigning vertices to the vehicles as long as its capacity is not exceeded. Then it tries to optimise each individual route separately by solving the corresponding TSP.

2.9.2 Heuristics for the VRPs

Typically, problems addressed as VRP are large in number and complex in the nature as presented in section 2.6. Researchers have specified different categories to describe VRPs. There are many heuristics developed to address different categories of VRPs. In general, the solution strategies implemented by these heuristics can be classified into five main groups as described in Table 2-1. Also,

Table 2-2, represents a number of these heuristics. Also Appendix A provides examples of the algorithms used to address each VRP category.

Table 2-1, Solution Strategies implemented by different VRP heuristics

Solution Strategy	Description	Examples
Cluster first-route second	This approach first groups/clusters demand nodes/arc and then, it designs economical routes over each cluster as a second step.	The Sweep Approach [36].
Route first-cluster second	First, a large route or cycle is constructed including all the demand entities. Next, the large route is partitioned into a number of smaller, but feasible routes.	Early works down by [37].
Saving/Insertion procedures	At each step, the current configuration of nodes is compared with an alternative configuration. The outcome alternative configuration with the largest saving in terms of some objective function is chosen.	Early works down by [38].
Improvement/ exchange procedures	This procedure always maintains feasibility and strives towards optimality. At each step, one feasible solution is altered to yield another feasible solution with the reduced overall cost.	Branch Exchange heuristic[29].
Mathematical programming	It is based on a mathematical programming formulation of the underlying routing problem.	[32,25, 26] used Zero-one integer program.

2.9.3 Heuristics for the VSP

The VSP and its specified categories VSPLRP, VSPMVT and VSPMD can be formulated as optimisation problems on appropriately defined networks. The use of the Minimum Cost Flow Algorithm to solve VSP was first suggested by Dantzing and Fulkerson [13]. In addition the Concurrent Schedule Method was proved successful in practice for solving a variety of constrained scheduling problems [39]. This heuristic is easy to code and computationally efficient and it is widely used in practice. Also, the Two Step Approach algorithm [6] was used to address VSPMVT and VSPMD. The output of these problems is a set of schedules clustered either by vehicle type or depot. This algorithm addresses these problems using the following two approaches:

- **Cluster tasks and then schedule vehicle over each cluster,**
- **Schedules vehicles and then clusters vehicle schedules.**

Furthermore [40] suggested an Interchange Heuristic, assuming a starting solution is available, which is found by other approaches such as Concurrent Scheduler. This approach exchanges a section of individual schedules and tries to find a better schedule. This approach is a more global evaluation strategy and it allows a variety of complex cost functions to be examined.

Table 2-2, represents a number of these heuristics. Also Appendix A provides further detail on these algorithms.

2.9.4 Heuristics for the VRSPs

The combined VRPs and VSPs incorporate both the spatial and time window constraints of the real system. As a result, the nature of the problem would be even more complex and difficult to address. Primary algorithms used to address VRSP are such as "Route First" algorithm and the "Greedy Insertion" procedure [41]. The Route First algorithm uses the same principle as Route-First strategy described in section 2.9.2. In the greedy insertion approach, a route is constructed in an iterative fashion by adding demand arcs representing origin-destination pairs to the route one pair at a time. This procedure uses the ratio of marginal increase in savings divided by marginal increase in route time as a criterion for adding a demand arc.

As it is evident from the past literatures [6, 5, 42], up until 1980s very little works were done in considering time window constraints in the VRPs. Early works on this field primarily involved some case studies. Also approaches based on Dynamic programming and branch and bound techniques we introduced later [42]. Solomon et al. [43] extended VRP heuristics to incorporate the time dimension in the heuristic process. These heuristics are known as: the saving heuristic, a time-oriented nearest-neighbour heuristic, the insertion heuristics and a time-oriented sweep heuristic. Appendix A, suggests further details on these algorithms.

Table 2-2 Classical heuristics used to address VRSPs

Problems	No.	Type	No.	Categories	No.	Algorithms			No.	Algorithms		
Vehicle Routing Problem (VRP)	1	Separate-Single Origin & Destination points	1-1	The Shortest Path Problem	1-1-1	Dijkstai Algorithm						
					1-1-3	Multoterminal Shortest-Chain Route procedure						
			1-2	Shortest-Path Models with Fixed Charges	1-2-1	The modfied shortest Path Interative Procedure						
			1-3	The K-Shortest Path Problem	1-3-1	The Double Sweep Method						
			1-4	Minimal Spanning Tree Problem	1-4-1	Minimal Spanning Tree procedure						
	2	Multiple Origin & Destination points	2-1	Transportation Problem	2-1-1	Simplex Method	2-1-2	Hungarian Algorithm				
					2-1-3	Vogel Approximation Method (VAM)	2-1-4	The Northwest Corner Method				
	3	Coincident Origin and Destination points	3-1	The Travelling Salesman Problem (TSP)	3-1-1	Nearest Neighbour Algorithm	3-1-6	Insertion procedure	3-1-6-1	Nearest Insertion	3-1-6-2	Cheapest Insertion
					3-1-2	Nearest Merger Algorithm			3-1-6-3	Arbitary Insertion	3-1-6-4	Farthest Insertion
					3-1-3	Christofieds heuristic			3-1-6-5	Quick Insertion	3-1-6-6	Convex Hall Insertion
					3-1-4	Clark & Wright Saving Method			3-1-6-7	Greatest Angle Insertion	3-1-6-8	Difference X ratio Insertion
					3-1-5	Minimal Spanning Tree approach						
			3-2	Chinese postman problem	3-2-1	CPP Algorithm	3-2-2	Dynamic Programming approach				
			3-3	The Single Depot Multiple Vehicle Routing Problem	3-3-1	The Saving Algorithm	3-3-2	The Sweep Approach				
					3-3-3	The Penalty Algorithm	3-3-4	M-Tour Approach				
					3-3-5	A Generalised Assignment Heuristics [219]						
			3-4	The Multiple Depot Multiple Vehicle Routing Problem	3-4-1	The Assignemt-Sweep Approach	3-4-2	The Multi depot Saving Approach				
					3-4-3	The Multi depot Saving Approach for large Problems						
Vehicle Scheduling Problem (VSP)			1	Vehicle Scheduling Problem with Length of Path Restrictions (VSPLPR)	1,2,3	The concurent Scheduler						
			2	Vehicle Scheduling Problem with Multiple Vehicle Types (VSPMVT)	1,2,3	Two Step Approach Algorithm						
			3	Vehicle Scheduling Problem with Multiple Depots (VSPMD):	1,2,3	An Interchange Heuristic						
Vehicle Routing & Scheduling Problems (VRSP)			1	Dial-a-ride routing and scheduling Problems	1-1	The Saving Heuristic with Time Windows	1-2	A time-Orineted Sweep Heuristic				
					1-3	Insertion Heuristics	1-4	A Time-Orinented Nearest-Neighbour Heuristic				

2.9.5 VRSP Metaheuristics:

These heuristics were mainly developed during the last decade. These heuristics perform a deep exploration of the most promising regions of the solution space. [44] suggest that these methods typically combine sophisticated neighbourhood search rules, memory structures, and the recombination of solutions. The quality of solutions produced by these methods is usually much higher than that obtained by classical heuristics however, these methods require higher computation time. In addition, these procedures are usually context dependent and require finely tuned parameters, which may make their extension to the other solutions difficult. In a sense, metaheuristics are sophisticated improvement procedures and they may be thought of as further improvements to classical heuristics.

In addition [44] highlights the need of today's commercial packages for development of faster, simpler and more robust metaheuristics. To this end they presented a number of Tabu Search (TS) based algorithms to address VRPs. These are Taburoute, Taillard's algorithms , Xu and Kelly's Algorithm , Rego and Roucarios algorithm, the adaptive memory procedure of Rochat and Taillard and the Granular Tabu Search (GTS) of Toth and Vigo. The adaptive memory procedure of Rochat and Tillard is one of the most interesting developments in the area of Tabu search using the most recent concept of adaptive memory, which provides a pool of good solutions that is dynamically updated throughout the search process [44]. Also, GTS is yet another very promising concept. Toth and Vigo [45] showed that GTS produces excellent results within very short computing time.

Furthermore, [43] described a variety of route construction and scheduling heuristics that used time windows, typically known as VRSTW (i.e. Vehicle Routing and Scheduling with Time Windows). Also, [43] identified that a sequential time space based insertion algorithm outperformed the other approaches in VRPs. However [46] suggested that the weakness of this approach lies in the fact that in some cases the last un-routed customers tend to be widely dispersed over the geographical area, resulting in last routes with poor quality. For this reason they proposed a new parallel approach for initialising and building a set of routes. Further, they suggested that the parallel approach was found to be better than the sequential approach of [43] for pure random problems resembling real life situations. However, the [43] approach outperformed the parallel approach on pure clustered problems.

In addition, [47] further extend VRSTW to incorporate backhaul concept, these problems are referred to as VRSBTW (i.e. Vehicle Routing and Scheduling with Backhaul Time Window). They suggest a TS heuristic that addresses two different objectives. The first objective was to minimise the total distance travelled by vehicles and the second objective was to minimise the total route time. The TS algorithm was found to be effective on the VRPBTW, however the introduction of backhauls resulted in an increase in the number of routes and total route time.

Furthermore, [42] offered a two-stage optimisation approach, which is a combination of sequential and parallel insertion algorithms, followed by a Tabu Search algorithm. Also, [48] used a heuristic, with a parallel insertion algorithm and local search interchange, to develop a two-stage optimisation algorithm. Furthermore, [42] developed a two-stage approach using parallel insertion and interchange methods.

A number of researchers [3] suggested that the heuristic approaches were capable only to specific variation, a slight difference in the structure of the problem made the algorithm inefficient. This is mainly due to the mathematical structure of these approaches, which presents a number of disadvantages [50]:

1. It is difficult to formulate a reasonable objective function and hard constraints.
2. Mathematical formulation leads to a very large number of variables if all possible constraints are considered.
3. The optimum solutions obtained from these approaches are no longer valid in the case of unexpected events such as bad weather conditions, breakdowns and etc.

Therefore, these shortfalls persuaded researchers towards the development of more generalised approaches that were versatile enough to include more variations of the problem. The followings briefly introduces different approaches taken to address these problems

2.10 Enabling Technologies:

The advancement of computer and information technologies in the last decade has had a great influence on developing decision support systems, which incorporate operation research models and optimisation techniques. [25] reports the development of a number of decision support systems incorporating routing and scheduling heuristics such as:

- A DSS to support optimisation or routes and scheduling of freight operations was developed by incorporating a combination of heuristics and integer programming.
- The use of a cluster-first-route-second heuristic in a DSS to support delivery of orders to customers and to service customers.
- The use of a tour construction/improvement heuristic to support decision making for picking and delivering products to dairies.
- The use of sweep algorithm in a DSS for a diary company's vehicle routing and scheduling.

As a result of operations research interest "VERSA" and "TRAVELLER" were developed as packages to facilitate routing and scheduling orders from more than one depot. These packages were developed by the Imperial College London University and Leeds University and also the "PATH-FINDER" and "VAN PLAN" are standard packages offered by IBM [41].

In addition, the advanced IT systems could also be used to support the planning and management of distribution operations. To this effect [25] have developed a new DSS including a combination of Supply Chain Management (SCM) applications and a Graphical Information System (GIS) integrated with an Enterprise Resource Planning (ERP) software.

Also, to overcome the shortfalls of algorithmic methods in addressing VRSP problems, researchers [6, 51, 52]] have suggested developing interactive procedures where humans could play some part in the routing and scheduling process. In this way human expertise could lead to improvements or could transform unacceptable solutions into ones, which could be used in practice [36]. Furthermore, [33] argues that the conventional approaches do not reflect the knowledge and intelligence of a dispatcher, that is the ability to deal with uncertainty and imprecision. They further propose the development of DSSs with the ability to simulate the dispatcher's behaviour using Neural Networks.

Finally, it is evident from the current conventional approaches that they fail to consider practical requirements such as capturing the stochastic and dynamic behaviour of the system, considering several criteria and producing different alternative solutions. An improvement in capturing systems stochastic behaviours could greatly impact the credibility and reliability of the decision support systems. In addition, Multi-criteria analysis of the system would provide a major advantage in taking into account a range of different criteria, to assist a decision maker with better trade offs in addressing VRSP. Finally, providing several optimum or close to optimum solutions would assist decision makers to examine different alternatives for better assurance. The application of Artificial Intelligence techniques and simulation modelling concepts could be useful as addressed in the following sections.

2.11 Artificial Intelligence (AI):

Recently many researchers have proposed the use of AI techniques such as Expert Systems (ES), Neural Networks (NNs), Fuzzy Logic and Genetic Algorithm (GA) to address VRSP. To this effect [53] suggest that AI provides more flexible and expressive power than mathematical programming in modelling complex domain problems. Also, [50] Indicated that real problem complexities could not be included in existing algorithmic approaches to VRSP. They refer to studies performed in identifying characteristics that are best addressed by ES. Some of these characteristics are in common with VRSP, which are:

- Very complex that cannot be solved by an algorithm,
- Requires a large amount of data analysis,
- Not clearly defined with specific boundaries,
- Have recognised experts, who are better than armatures.

In addition, [27] suggest that in a scheduling problem with non-quantifiable situations and/or soft constraints, ES methods are found being more effective than the general optimisation methods. They further argue that in global optimisation methods, there is a lack of explanation for decision on assignments. Even though the ES do not have knowledge of the global optimum, they would increase the smartness and knowledge on making decisions. For this reason they proposed development of an ES for dynamic VSP.

Also, [54] were first to demonstrate the application of Neural Networks (NNs) in NP-Hard combinatorial problems. They introduced the well-known Hopfield NN to address TSP. Also, there have been studies in using NNs in fields such as VRPs, shortest path problem and scheduling applications. VRP has been addressed using self-organising NN [55]. In addition, [56] has used a Hopfield NN to locate an optimal solution for the shortest path problem. Furthermore, a stochastic NN approach was proposed to address resource constrained scheduling and project scheduling [57]. Also [58] proposed an alternative NN model for modelling the decision process of expert dispatchers for dynamic Dial-a-Ride problems. They concluded that a NN based expert dispatch system is feasible. They also emphasised the flexibility of NNs approach as an attractive feature allowing NN to be trained in various dispatching environments and dynamically adapt to them during their training phase. This adaptive behaviour is certainly a major asset, as compared with classical expert system approach where the decision rules must be “a priori” determined and tailored to each specific context.

Fuzzy logic was also used successfully to model situations in which people make decisions in an environment that is complex and very hard to develop a mathematical model [59]. In most vehicle routing models, it is assumed that travel time, transport costs and the distance between pairs of nodes in the network are constant values known in advance. However, dispatcher decision makers most often make a subjective estimate of travel time based on their experience and intuition [3]. For instance, a dispatcher typically expresses the estimated travel time as short, long and etc. [60] treated travel time and transportation costs between two nodes in network as fuzzy numbers. They modified the Clark and Wright [38] algorithm when travel times in a network are treated as fuzzy numbers. Also, [61] developed an algorithm to route vehicles when demand at the nodes is uncertain. This method is based on the Sweep heuristic and fuzzy logic.

In order to overcome the real situation in which there exists vagueness and difficulty determining the amount of supply and demand as specific numbers, [62] considered a fuzzy version of the transportation problem by introducing two kinds of membership functions which characterize fuzzy supplies and fuzzy demands. The objective was to determine an optimal flow that maximizes the smallest value of all membership functions under the constraint that the total transportation cost must not exceed a certain upper limit.

Also, [63] showed that route choice in vehicle routing and scheduling plays a critical role in many transportation problems. The route choice is typically characterized by subjectivity, ambiguity and uncertainty. In such cases, fuzzy logic seems to be an appropriate approach to deal with these concepts. In addition they developed a heuristic way for handling fuzzy perceptions in explaining route choice behaviors. They developed a hybrid model where route choice decision-making is described in a hierarchy using concepts from fuzzy logic. Also the Analytical Hierarchy Process (AHP) is proposed for making possible a more proper description of route choice behavior in transportation systems.

GAs as introduced in 2.3.5.3, are a member of evolutionary algorithms, which is a family of computer models based on natural selection and natural genetics. GAs are typically used for optimization problems. These algorithms are applied highly to linear, non-linear, stochastic combinatorial complex problems, where it is hard to find a model, which provides an approach to the solution. However, GA is not guaranteed to find the global optimum of a problem, it only ensures to find better solutions than randomly initialized ones. GAs represents a number of features providing better efficiency and robustness and more advantages than other searching methods. These features include [64]:

- GA deals with a parameter code and not the parameters themselves,
- GAs are multiple parameter searching algorithms. They search multiple combinations of different parameters. Thus, the probability of obtaining local optima would be reduced.
- GA uses objective function information only, and they do not use any other auxiliary mathematical knowledge, such as gradients, derivatives, etc. therefore, GAs may have wider ranges of applications than mathematical based methods.
- GAs exploit the principles of rules for probabilistic transition to find optimal solutions.

These features provided considerable flexibility in adapting this technique to particular applications. For this reason these algorithms are being considered for use in those problems with complex solution spaces for which there are not efficient algorithms. A number of GA application to address VRSP are presented in Appendix A. Also a more in depth review of GA in VRSP is provided in chapter 3.

In addition to the above applications, GAs are powerful optimisation tool, which are particularly appropriate to multi-objective optimisation. In classical methods, an algorithm has to be applied many times to find multiple Pareto-optimal solutions. In addition some algorithms are sensitive to the shape of the Pareto optimal front [65]. However, GAs maintain a population of solutions and therefore, can search for many non-inferior solutions in parallel. The GAs ability to produce a set of solutions in a single run without converging to strong domain specific assumptions provides benefits over conventional Multi-objective methods. To this effect, [66, 67] used GA to address multi objective transportation problems. Their works are briefly described in Appendix A.

2.12 Simulation Technique

Simulation is a modelling technique that is gaining popularity in tackling problems faced by manufacturing and service industries. A number of researchers [68] define simulation as the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and or to evaluate various strategies for the operation of the system. This technique can then assist the user in decision making, thereby reducing the time and costs associated with experiments on a real system. This also minimises risks associated with experiments on real systems.

However, simulation as an experimental tool does not solve a problem or optimise a design. It supports in evaluating a solution and providing understanding of problematic areas rather than generating a solution. An optimum solution can only be obtained through experimentation by running and comparing the results of alternative solutions.

Simulation is a powerful, versatile, and flexible tool that is being used in different ways through a variety of manufacturing and service industries. In particular simulation modelling could be useful for resource scheduling. Scheduling can be described as the allocation of available resources over time to meet performance measures defined for a system. In terms of scheduling theory, most of scheduling problems are in the class of NP hard.

Simulation tools provide a set of rules and definitions that allow one to define a scheduling system in term of its constraints and the way in which resources are used to perform operations. By setting experiments to examine effects of different combinations of resource assignments on the overall systems performance, a feasible schedule may be obtained. In this way, the scheduler can be sure that all the possible combinations and exceptions are considered and the systems' objectives are met. Therefore, simulation can provide management with data that can assist in more informed decision making for scheduling and utilisation of the system resources. Also, [69] suggested that both simulation and optimisation models are needed to meet the changes in transportation and logistics/supply chain problems of today and the future. For this reason they further suggest that the simulation packages need to incorporate logistics and transportation constructs and terminology to facilitate model building in a more convenient way.

Also [70] developed an agent-based modelling simulation approach to assist management of logistics operations. This approach was conducted due to the need for mangers of logistics operations to be able to adapt the ever-changing demands of their environment and customer demands. The application and benefits of simulation modelling in logistics and supply chain management are further described in chapter 3.

2.13 Conclusions

This chapter highlighted Logistics management as being evermore-complex processes due to greater demands from customers and increased competition from competitors. One area that determines the efficiency of logistics management is the Vehicle Routing and Scheduling (VRS) activities. The complexity of these problems is mainly due to the existence of many parameters and constraints concerning manpower, vehicles, environments and other real world constraints.

The VRSP is formulated as a NP-Hard combinatorial problem, which requires the optimum allocation of resources to manage required transportation with respect to the operational constraints and time restrictions. To this effect a review of optimisation methods such as integer and dynamic programming were provided. However, both IP and DP require large data storage and computation time when faced with complex real world problems. Heuristics as an alternative approach were introduced. These methods typically require reasonable computation time to generate solutions close to the optimum. Also, some widely used general-purpose heuristics to address combinatorial optimisation problems were introduced. Some of these methods were such as tabu search, simulated annealing and Genetic algorithms. Although each of these methods presents potential advantages but GA represents a number of features that would result in a better efficiency and robustness, making this technique more flexible and adaptable for particular applications.

This chapter reviewed some of the existing methods used to address VRSPs. It was evident that the existing approaches are found inadequate in addressing VRSPs. Some of the clear shortfalls of these methods are such as [3, 71]:

- The traditional optimisation approaches have very limited applications and these require high computation time.
- The heuristic approaches are capable only to specific variation, a slight difference in the structure of the problem make the algorithm inefficient.
- The metaheuristics methods require higher computation time and they are context dependent.

Any improvements in addressing the above shortfalls could greatly impact the credibility and reliability in managing VRS activities. To this effect, this chapter presented a number of enabling techniques such as the use of AI and simulation techniques. These techniques present potential benefits in addressing the current shortfalls. The next chapter further illustrates how these techniques could be used to assist in addressing VRSP.

Chapter 3: Methodology

3.1 Introduction

It was evident from literature provided in the last chapter that the management of logistics activities is found to be a difficult and complex task. This could be partially due to factors such as uncertainty, vagueness, subjectivity and non-quantifiable measures that managers are faced with. For example uncertainty exists in the availability of raw material at supply sources and demand levels for the final items by the customers. Also decision on route assignments is not normally based on quantifiable and specific metrics. Parameters such as convenience and safety could play an important role in route assignment, which are not clearly defined or addressed. Above all, logistics operations are preformed in a least automated and controllable environment. In this environment, the logistics activities are subjected to uncertain road and weather conditions, geographical restrictions and resources availability leading to much subjective decision-making.

The conventional algorithmic approaches were shown to be inadequate in capturing and therefore addressing the real world complexities. In particular these methods failed to capture the stochastic and dynamic nature of the VRSP, they did not address the multi-criteria nature of these problems and also these failed to provide alternative solutions. Improving any of these mentioned areas would greatly impact the flexibility and credibility of decision-making in handling practical VRS problems.

For this reason, this chapter introduces an architectural design for a hybrid decision support system to address vehicle routing and scheduling for more flexible and versatile logistics management. Each component of this architecture is further described. The principal aim of this chapter is to introduce search the engine component of the proposed DSS. For this reason it describes the concept of random search; genetic algorithm and Pareto genetic algorithm. Furthermore, this chapter provides a review of the different Genetic formulations of vehicle routing and scheduling problems. It further describes the recent developments in field of Genetic algorithms.

3.2 Proposed Architectural Design

A decision support system is proposed as part of requirements to assist managements in decision-making in terms of finding an optimum or near optimum monthly LPG transportation schedule. The proposed DSS is composed of two integrated modules, a search engine and a simulation module as shown in Figure 3-1. The following briefly specifies the functionality of each of these modules.

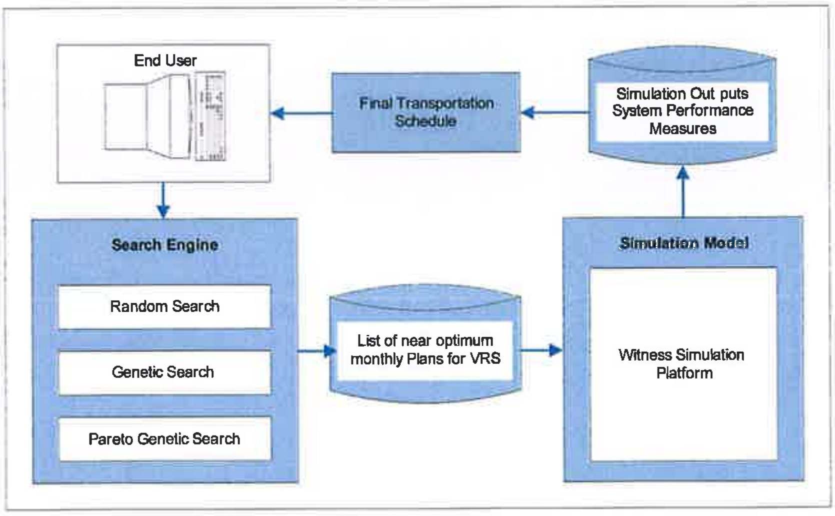


Figure 3-1DSS Architectural Design

- **The search Engine Module:**

The purpose of this module is to search for the near optimum schedules for LPG transportation. A schedule specifies the fleet size, the routes to be taken by each truck considered and also it specifies the expected loading and unloading times for transportation by trucks. There were three different search engines developed in this platform. These search mechanisms are based on random search, Genetic Algorithm and Pareto Genetic algorithm, which are further described in latter sections. Using any of these options a large candidate transportation schedule is generated. The outcome of this search engine is a list of optimum transportation schedules, which are stored in database for latter applications. This module of the DSS is further described in this chapter.

- **The Simulation Module:**

Witness simulation tool was used to develop a simulation model from the distribution system under investigation. This module uses the data in the database to generate a simulation model for the system. The simulation model allows for the incorporating of the dynamic behaviour of the system in terms of real constraints such as: refineries, processing plants and vehicle breakdowns, availability of resources, supply and demand levels and etc.

Also, using the transportation schedules from the search engine, the simulation model is used to evaluate the effect of different vehicle schedules on the systems performance measures such as average delays, deliveries completion times, work in progress, inventory levels and etc. The output from this module is stored in a Database for any further analysis. The simulation module covered in chapter 4.

3.3 Random Search

The random search technique is the most well known and also the simplest of stochastic methods as introduced in chapter 2. This method depends on randomly searching the objective function parameter space in order to find the optimum value of the objective function. As indicated in [72], this method is a powerful approach due to its lack of necessity of differentiating the objective function and also its simplicity. This simplicity makes it clear and customisable for various special problems.

In this approach a random search technique was used to randomly search the solution space for the optimal transportation schedule. In this method transportation schedules are generated randomly. That is the decision variables such as: trucks, routes and start times are randomly selected from their respective decision space. Based on these decision variables transportation schedules are generated and the objective function is evaluated to find the fitness of the schedule. This method unlike the genetic algorithm does not introduce any intelligence or stochastic operations on selecting different decision variables. The main aim in developing this method was to compare the performance of Genetic search with the random search for validation of the Genetic algorithm developed for the considering problem.

3.4 Genetic algorithms (GA)

Genetic algorithms have been developed by John Holland [73] at the University of Michigan during the 1970's. Goldberg [74] describes Genetic algorithms as search procedures based on mechanics of natural selection and natural genetics. The Literature refers to his original work as the "Simple Genetic algorithm " (SGA), which is used as a foundation of almost any new work carried out in this area.

In its most general form, a SGA creates a population of feasible solutions and generate future populations based on natural selection. To complete an optimisation problem by GA, there are four steps that should be followed.

- The first one is to represent the problem with a bit string (chromosome) so that its optimisation can be performed based on it.
- The second step is to produce an initial population of offspring generated randomly.
- The third step is to compute the fitness of each member of the population by an evaluation function.
- Then given a population of individuals, a number of individuals are selected and changed by means of crossover and mutation to produce new offspring.

The probability of a particular individual passing its genes into a succeeding population is directly proportional to its fitness. The fitness of a string is a measure of the quality of the solution represented by the string. The higher fitness corresponds to a high value or a low value of the function considered. The fitness evaluation and operations by crossover and mutation operators are very important and determine the evolution level and convergence speed. In the SGA although the selection is clearly not "Natural", individuals are selected to survive into the next generation on the basis of their fitness. In SGA each generation produces the next and then dies off, so that an individual in generation $\boldsymbol{g}$ never has a chance to breed with one in generation $\boldsymbol{g+1}$. Figure 3-2, illustrates the working of the SGA and its basic components.

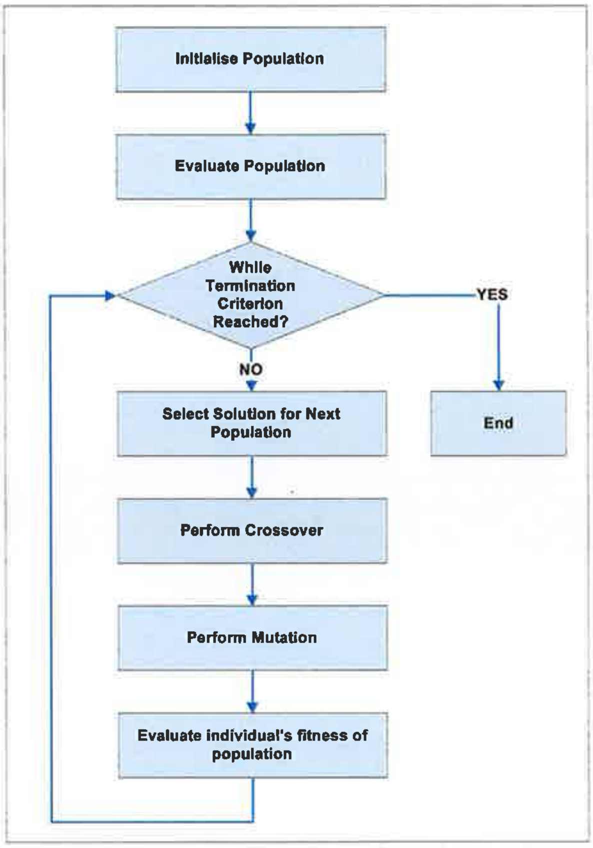


Figure 3-2 Format of a Simple Genetic algorithm

In each cycle of GA operations, a new generation of possible solutions for a given problem is produced. After some number of generations, it is hoped that the algorithm will converge on the best individual in the final population representing a near optimal solution [75].

GAs have been quiet successfully applied to discrete optimisation problems like routing, scheduling, transportation problems, travelling salesman problems, etc. However, [76] warned against perceiving GAs themselves as optimisation algorithms. He suggests thinking about GAs as highly idealised simulations of natural process, in which they embody the goals and purposes of the natural process.

In general GA has eight basic components: genetic representation, Initial population, evaluation function, reproduction, selection scheme, genetic operators, stopping criteria and GA parameter settings. These parameters are further described in the following sections.

3.4.1 Genetic Representation for potential solutions

In GAs, chromosomes must be encoded in a way to represent a complete solution for the considered problem. As indicated earlier the classical representation for GAs is binary digits. However, the encoding mechanism highly depends on the nature of the problem variables. In fact, other representations such as integer and floating point have been found to find better solutions for different problems. Reeves [77] suggests, one of the main reasons for GA failures is due to selection of inappropriate coding scheme.

3.4.2 Generation of Initial Population

The generation of the initial population is necessary to start the GA process for finding the optimum solution. The size of the initial populations is an important parameter to be specified. The size refers to the number of chromosomes that are in a population. If there are too few chromosomes, then the GA have few possibilities to perform crossover and therefore only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA slows down. For the given size, chromosomes are often constructed randomly by sampling values for variables in their respective decision spaces. The generated chromosomes may not satisfy all the constraints considered in a problem. Therefore, the aim of the GAs process would be to improve the individual chromosomes of the population through successive generations to find the optimum chromosome (i.e. solution).

3.4.3 Fitness Function and Evaluation of Chromosomes

The generated chromosomes are evaluated based on their fitness. Each individual chromosome represents a potential solution to a problem. The evaluation function assigns a real number as a measure of fitness to each solution. This function is used to rate solutions in terms of their fitness. Usually the evaluation function is a monotonic function of the problem objective function. The performance of each chromosome is then reported with respect to the constraints imposed by the problem.

3.4.4 Selection For Reproduction

One of the main GA operators that would influence the performance of the evolutionary algorithm is the selection method. Selection is the operation whereby candidate solutions (i.e. chromosomes) are selected for reproduction. In general the probability of selection should be proportional to the fitness of the chromosome in question.

The main goal of the selection procedure is to reproduce more copies of chromosomes whose fitness values are higher. In a GA, the selection is based on the natural law of the survival of the fittest among the chromosomes. It has a significant influence on driving the search towards a promising area and finding good solutions.

Whitley [78] argues that there are only two primary factors in a genetic search, which are associated to selection schemes: *population diversity* and *selective pressure*. The *population diversity* is the portion of chromosomes of a population that is selected during the selection phase. The *selective pressure* is the probability of the best chromosome being selected compared to the average probability of selection of all chromosomes. These parameters have a great influence on the performance of the Genetic algorithm and a good selection scheme must have a balance between these two. As selective pressure is increased, the search focuses on the top individuals in the population, but because of this 'exploitation' genetic diversity would be lost and this may result in premature convergence. Reducing the selective pressure increases 'exploration' as there are more genotypes and thus more schemata involved in the search and consequently this may make the search slow and ineffective.

The first and possibly the most recognised work in this area were by De Jong [79] in 1975. Since then a number of selection schemes have been suggested for GA. These methods could be either deterministic or stochastic. Some of these techniques are described in the following sections:

3.4.4.1 Fitness Proportionate Selection

The most common approach is fitness proportionate selection. In this scheme, a string with fitness value f_i is allocated f_i / f_m offspring, where f_m is the average fitness value of the population. A string with a fitness value higher than the average has a chance of allocating more than one offspring, while a sting with a fitness value less than the average may allocate no offspring in the next generation. For this reason the allocation techniques usually include some randomisation to remove methodical allocation biases toward any particular set of strings to match the expected number of offspring f_i / f_m.

The SGA uses the roulette wheel selection scheme to implement proportionate selection. This selection method is in fact a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness value. The better the chromosomes are, the bigger their fitness value, the larger their section and therefore the more chances to be selected. This method chooses chromosomes probabilistically, instead of deterministically. In this way, if a chromosome has the highest fitness there would be no guarantee that it will be selected. In fact, on average a chromosome will be chosen with the probability proportional to its fitness.

One of the drawbacks of roulette wheel selection is that highly fit chromosomes would take a significant portion of the wheel resulting in premature convergence, when there is insufficient difference between solutions.

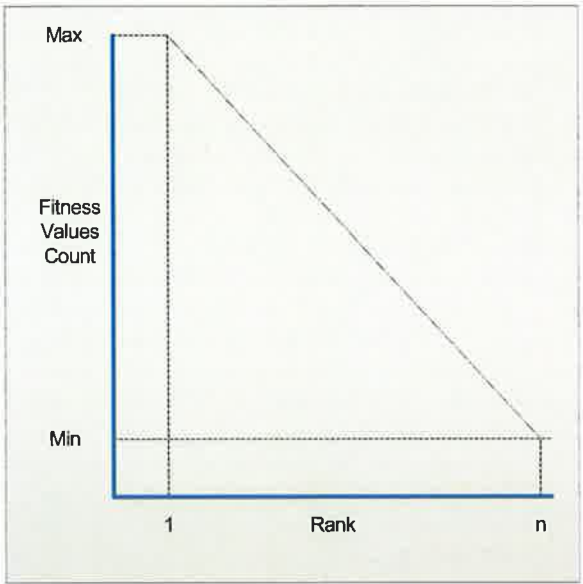


Figure 3-3 Fitness assignment in the ranked based Algorithm Scheme [74]

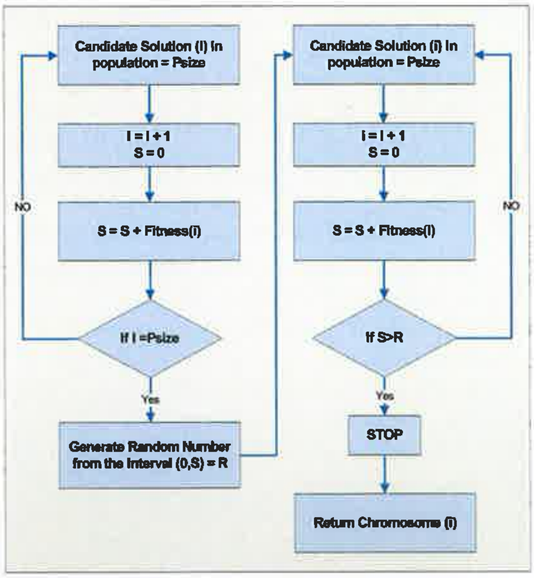


Figure 3-4 Roulette Wheel Selection

3.4.4.2 Tournament Selection

Tournament selection [80] could be considered as a noisy version of the rank selection. In a single iteration, this method randomly selects some number k of individuals and selects the best one from this set of k elements into the next generation. This process is repeated for population size a number of times. Typically the k value is set to 2. It is possible to use either raw or scaled fitness with this scheme. This process inherits the advantages of rank selection but does not require the global reordering of the population and it is more inspired by nature.

In addition to the above-motioned methods, Appendix B provides description on the Rank-Based Selection method. [81] compares four different schemes; proportionate selection, fitness ranking, tournament selection and steady state selection. They concluded that by suitable adjustment of parameters, all these schemes apart from proportionate selection can be made to give similar performances, so there is no absolute best method.

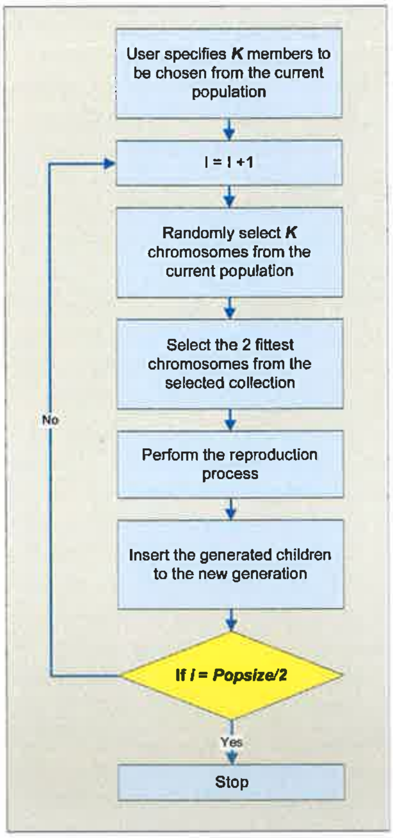


Figure 3-5 Tournament Selection Method

3.4.5 Recombination Process Using Genetic Operators

The Genetic algorithm could be described as proceeding from one population to another. This is achieved by recombination process, which uses genetic operators to produce a new population of individuals by manipulating the genes possessed by parents of the current population. This operation involves two operations known as crossover and mutation.

3.4.5.1 Crossover

Crossover serves as a mechanism by which parent chromosomes can exchange information (genes) and possibly create more fit new offspring and therefore allowing the exploration of new regions of the search space. The idea behind crossover is that the new chromosomes may be better than both of the parents if it takes the best characteristics from each parent. This is an operator that enables the GA to exploit the current neighbourhood and is expected to move the GA to a local optimum.

In general crossover takes place during evolution based on the user defined crossover probability (P_c). The value of P_c lies in the range of 0 to 1. If $P_c = 0$, then there is no crossover and the population is made from exact copies of chromosomes from the old population. When $P_c > 0$, a part of the new population is formed by crossover and if the crossover probability is one, then all the new offspring are made by crossover. There are a number of ways to exchange these genes. A number of these methods are presented here. The reader can refer to Appendix B for further reading on Order and Cycle Crossover operations.

The Classic Crossover: Holland [73], has presented two Classic Crossover techniques known as The Single Point Crossover and Two-Point Crossover. Most of the new approaches developed mainly involve variations of either of these operators.

- **Single Point Crossover:** This operator was first introduced by Frantz [82]. Considering that l is the string length of a chromosome, a single point crossover is randomly chosen in the range of 1 and l-1. The portions of the two strings beyond this crossover point are exchanged to form two new offspring. The crossover point may assume any of the l-1 possible values with equal probability. Figure 3-6 and Figure 3-7 illustrate this process.

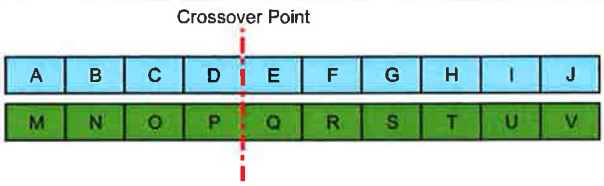


Figure 3-6 Before Crossover

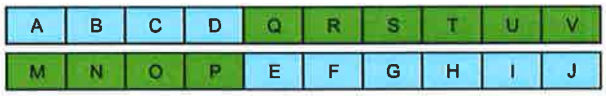


Figure 3-7 After Crossover

- **<u>Two point Crossover:</u>** This form of crossover was first defined by Cavicchio [83]. In this operation, two crossover points are randomly chosen. The strings between successive crossover points are swapped between the two parents to generate two new offspring.

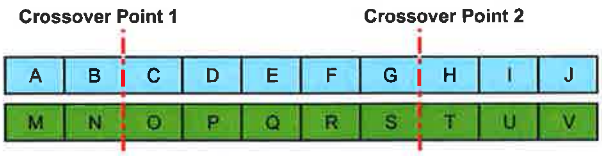


Figure 3-8 Before Crossover

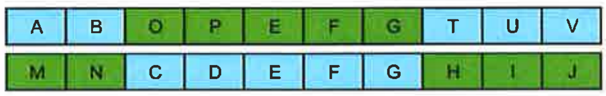


Figure 3-9 After Crossover

In addition to the classic genetic operators, a number of crossover techniques have been developed which are mainly used for route representation. These operators are mainly used to address route assignments in TSP. These operators are as follows:

- **<u>Partially Mapped Crossover (PMX):</u>** This operator was proposed by Goldberg and Lingle [84]. Using this operator, two strings are aligned and two crossing sites are picked uniformly at random points along the strings. These two points define a matching section that is used to affect a cross through position-by-position exchange operations. For example the two parents with the two cut points marked as indicated in Figure 3-10 would produce offspring in the following way:
- First the segments between cut points are swapped as shown in Figure 3-11, (the symbol X represents the unspecified genes).
- This swap also represents a series of mappings as indicated in Figure 3-12.
- Then fill the genes from the original parents, for which there are no conflict as indicated in Figure 3-13.

- Finally, replace the X locations using the mapping relationship exists as shown in Figure 3-14.

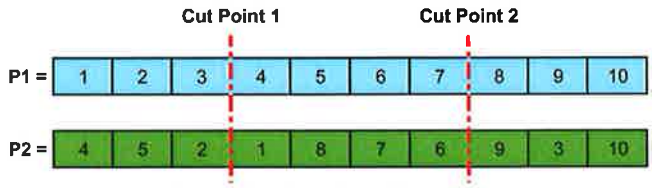


Figure 3-10 Before Mapping

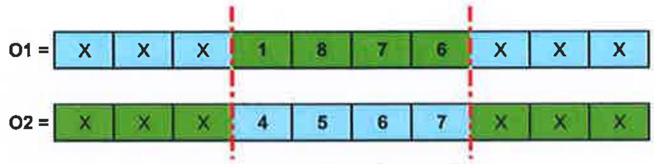


Figure 3-11 After Swapping

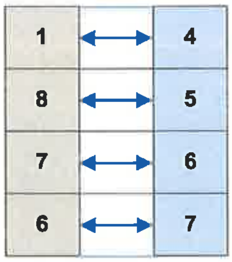


Figure 3-12 Mapping strings

Figure 3-13 Replacing genes with no mapping conflicts

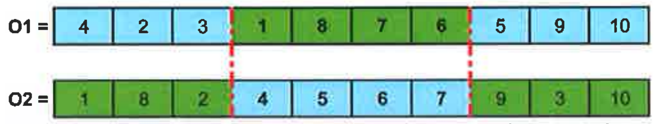


Figure 3-14 Replacing genes using the mapping relations

The PMX is aimed at maintaining the inheritance of adjacency and the relative order of the genes in the chromosome structure.

3.4.5.2 Mutation

After crossover, strings are subject to mutation. This is a genetic operator that alters one or more genes of a selected chromosome from their initial state with a probability equal to the mutation rate. The idea behind this operator is to increase diversity in a population.

Mutation probability (P_m) defines the probability of mutation of chromosome. It must be noted that the genes are independently mutated and therefore the mutation of a gene does not affect the probability of mutation of other genes. Mutation is considered as a secondary operator with the role of restoring lost genetic material. For example, if all the string in a population have converged to a 0 at a given position. Then crossover cannot regenerate a 1 at that position, while a mutation could. In general mutation is used to prevent a genetic algorithm falling into local extrema, but it should not occur regularly, because then the GA would in fact change into a random walk. The mutation may result in finding better optima.

Sometimes, especially late into the GA run, populations can stagnate and concentrate around local optima. When this happens, the gene pool can become too concentrated and standard mutation rate cannot generate sufficient diversity to enable the algorithm to free itself quickly enough. To overcome this, the mutation rate is raised to a higher level for a generation or two. This process is called *hyper-mutation.* A number of mutation techniques are available, which are described as follows:

Classic Mutation: this is also known as Flip Bit. In binary coding the flip bit involves random alteration of the value of a string position. This simply means changing a 1 to 0 and vice versa as indicated Figure 3-15.

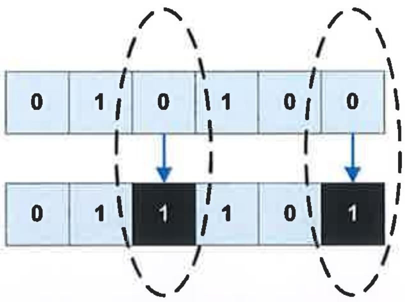
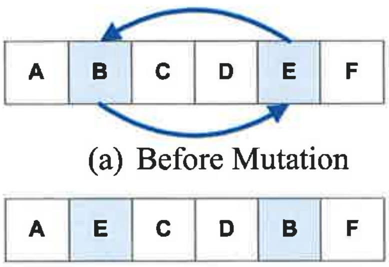


Figure 3-15 Flip Bite Mutation

Swap Mutation: This is the smallest possible random modification of a chromosome. According to the mutation probability, one or more pairs of genes are selected randomly and they are swapped to produce new offspring. Figure 3-16 illustrates this process, where genes B and E are randomly selected and they are swapped to produce the new chromosome.

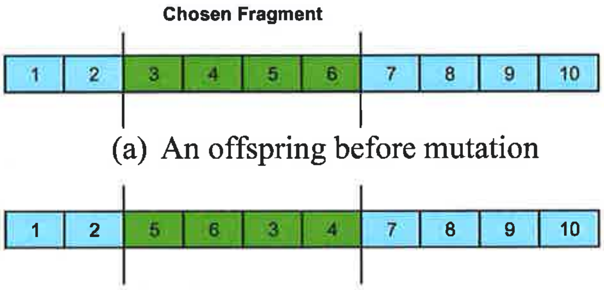


(a) Before Mutation

(b) After Mutation

Figure 3-16 Swap Mutation

Scramble Mutation: Davis [85] introduced this operator as an alternative to the swapping mutation. He suggested that the swapping method does not work as well as expected having tried it on many different problems. The scramble method selects a fragment of genes from a chromosome and scrambles it in the offspring as shown in Figure 3-17.



(a) An offspring before mutation

(b) After scrambled mutation

Figure 3-17 Scrambled mutation

3.4.6 Generational Cycle

The generational cycle in GA is the cycle of reproduction of a new population using a selection scheme, and recombination of chromosomes using GA operators (Crossover & Mutation). Figure 3-18 highlights the generational cycle within GA operations.

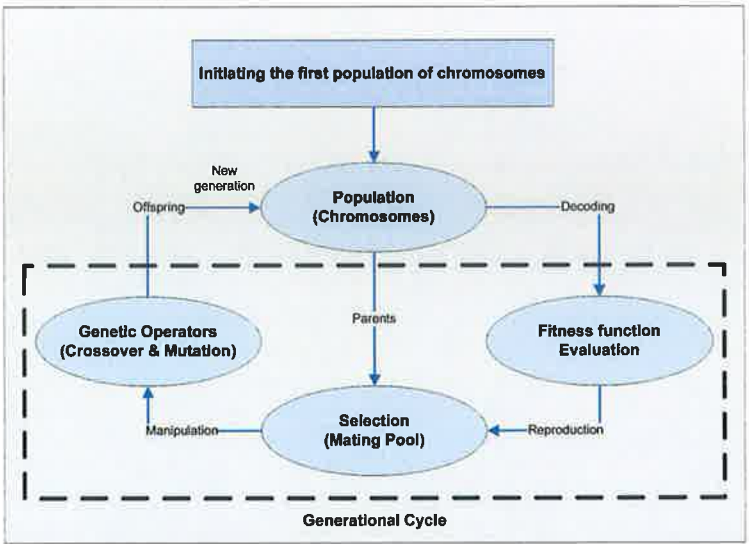


Figure 3-18 GA Generational Cycle

Figure 3-19, represents an example of a generational cycle with an initial population of four chromosomes of 10 genes each. The objective function, which is used to measure the goodness of each chromosome in the pool, is the summation of ones in a string. Then, objective function is divided by ten to normalise the fitness values to a range 0 to 1. As indicated the four chromosomes have fitness values of 0.5, 0.5, 0.2, and 0.9 respectively. Based on the proportional selection scheme, there must be 1.0, 1.0, 0.4, 1.7 offspring allocated to each chromosome. However, the final allocations of offspring in this case are: 1, 1, 0, and 2.

	1	2	3	4	5	6	7	8	9	10	Fitness Value	Total Fitness	Average Fitness
	Initial Population												
Ch1	0	0	0	0	1	1	1	1	1	0	0.5		
Ch2	1	0	0	0	1	0	0	1	1	1	0.5		
Ch3	0	1	0	0	1	0	0	0	0	0	0.2		
Ch4	1	1	1	1	1	1	1	1	1	0	0.9	2.1	0.525
	Selection												
Ch4	1	1	1	1	1	1	1	1	1	0	0.9		
Ch4	1	1	1	1	1	1	1	1	1	0	0.9		
Ch1	0	0	0	0	1	1	1	1	1	0	0.5		
Ch2	1	0	0	0	1	0	0	1	1	1	0.5	2.8	0.7
	Crossover												
Ofs1	1	1	1	1	1	0	0	1	1	1	0.8		
Ofs2	1	0	0	0	1	1	1	1	1	0	0.6		
Ofs3	0	0	1	1	1	1	1	1	1	0	0.7		
Ofs4			0	0								2.8	0.7
	Mutation												
Ofs1	1	1	1	1	1	0	0	1	1	1	0.8		
Ofs2	1	0	0	1	1	1	1	1	1	0	0.7		
Ofs3	0	0	1	1	1	1	1	1	1	0	0.7		
Ofs4	1	1	0	1	1	1	1	1	1	0	0.8	3	0.75

Figure 3-19 A generational cycle in SGA

The next step is to generate a second-generation pool of chromosomes, which is done using a selection operator, which results in the selection of chromosome 1, 2 and 4. Selection is biased towards elements of the initial generation, which have better fitness. Following selection, the crossover (or recombination) operation is performed upon the selected chromosomes. This involves randomly pairing individuals for crossover operation. As indicated in Figure 3-19, chromosomes 1 and 4 form one pair, while chromosomes 2 and 3 form the second pair. The crossover rate is assumed to be 100%. Therefore, in the first pair the crossover point falls between forth and the fifth bits of the strings and the portions of chromosomes 1 and 4 beyond the fourth gene are swapped. In the case of the second pair, the crossover point randomly falls between the second and the third genes. Population 3 represents the chromosomes after crossover.

The next step is to mutate the newly created offspring. The action of mutation on population 3 can be seen in population 4. Only two genes out of 40 have been mutated, which means an effective mutation rate of 0.05. These processes ultimately result in a second-generation pool of chromosomes that is different from the initial generation. Population 1 represents the initial population and population 4 represents the next generation. Populations 2 and 3 represent the intermediate stages in the generational cycle.

Generally, the average degree of fitness will be increased by this procedure for the second-generation pool, since only the best organisms from the first generation are selected for breeding. The entire process is repeated for this second generation. The process is repeated until a stopping criterion is reached. This could be terminated after a fixed number of generations, after a chromosome with a certain high fitness value is obtained, or after all the strings in the population have attained a certain degree of homogeneity.

Typical SGA uses a population size of 30 to 2000, crossover rates from 0.5 to 1.0 and mutation rates from 0.001 to 0.1. The population size, mutation rate and crossover rate are together referred to as the control parameters of the SGA and these must be specified before its execution.

A slight variant of this method of pool generation is to allow some of the better organisms from the first generation to carry over to the second, unaltered. This form of genetic algorithm is known as replacement strategy.

3.4.7 Replacement Strategies

After every crossover and mutation, new strings are created. Poor performing offspring are replaced in the new generation with a replacement strategy. These strategies specify how the next generation are to be created. Typically, the child replaces the parents. However, there are different variations to this rule attempting to preserve population diversity.

De Jong [79] suggests that most of these methods are instances of the **crowding methods**. In these methods, new individuals are more likely to replace existing individuals in the parent population that are similar to themselves based on genetic similarity. They have been used to locate and preserve multiple local optimums in multimodal problems. Effective crowding methods are such as: **Restricted Tournament Selection (RTS)** [86], **Deterministic Crowding** [87], **Elitist Recombination** [88] and **Keep-Best Reproduction** [89].

In addition, some replacement strategies developed for Steady State Genetic algorithms (SSGAs) include: **first-in-first-out strategy (FIFO)** [90] and **conservative strategy** [91]. In this study, the **ELITISM** strategy was implemented. In this method the best individual at generation k (the father) is maintained in the next generation $k+1$ if its child has a performance inferior than its father. Typically, the best individuals are placed in a temporary buffer before selection and then added into the new population after selection; crossover and mutation have been carried out. Elitism can very rapidly increase performance of GA, because it prevents losing the best-found solution to date. A variation to this method is to eliminate an equal number of the worst solutions that is for each "best chromosome" carried over a "worst chromosome" is deleted. Without the elitism, the best results can be lost during the selection, mutation and crossover operations. Some recent GAs methods implementing elitism approach are presented in [92, 93, 94].

3.4.8 Initialising the values of control parameters

GAs find solutions by exploration and exploitation of the search space. The functionality of GA is mainly based on its common balance on its exploration of new search regions and its exploitation of the already sampled region. This balance is highly dependent on the values of the algorithm's control parameters such as crossover probability (P_c), mutation

Probability (P_m), maximum number of generations (G_n) and number of individuals in the population (N). The Crossover and mutation operators are two important genetic operators, where crossover allows the algorithm to explore a particular hyper plane and mutation leads the algorithm to exploit different hyper planes. Therefore, the interaction of genetic operators is essential in reaching the optimum solution.

One of the earliest studies of genetic algorithm control parameters is that of [79]. Based on his observation, he suggested that a setting for population size, probability of mutation and crossover with values 5, 0.60 and 0.001 respectively, would give satisfactory performance over a wide range of problems.

Expanding on this work, [95] performed a more systematic study on the effects of the control parameters. They used a GA to optimise these parameters and concluded that the setting of (N=80, P_c=0.45, P_m= 0.01) would give to the best performance when considering the average fitness of each generation as the indicator.

While the choice of the initial values for the control parameters has been analysed in both analytical and empirical investigations, no theoretical method has been introduced to select the optimal control parameters for the GA yet [96]. However, the following factors should be considered in the selection of the initial value for the control parameters.

- Increasing the crossover probability increases recombination of chromosomes, but also it increases the disruption of good strings.
- Increasing the mutation probability tends to transform the genetic search into a random search, but it also helps to re-introduced lost genetic material.
- Increasing the population size increases its diversity and reduces the probability that the GA will prematurely converge to a local optimum, but it also, increases the time required for the population to converge to the optimal regions of the search space.
- The number of generations influences the searching time and searching result. The GAs with larger search space and less population size would need more generations for to converge to a global optimum.

Several researchers have proposed control parameter sets that guarantee good performance on chosen test-beds of objective functions. Two distinct parameter sets have been introduced: one has small population size and relatively large mutation and crossover probabilities, while the other has a large population size, but much smaller crossover and mutation probabilities. These methods are introduced in [97] and [98] respectively.

3.4.9 Comparison of GA with other Optimisation techniques

Several well known optimisation methods have been proposed for combinatorial optimisation problems such as Dynamic Programming (DP), Lagranean Relaxation (LR), and other heuristics as presented in chapter 2. Researchers suggest [99, 100] that DP is a time consuming algorithm, LR is limited by "duality gap" which will result in the loss of a global optimum solution. Also heuristics do not provide high quality solutions, and they need high computation time for better quality solutions. In addition, the Simulated Annealing (SA) and Tabu Search (TS) algorithms as were introduced in chapter 2. Studying the characteristics of each of these general optimisation methods, a number of facts could be summarised as follows:

- The focus of TS and SA has traditionally been that of combinatorial optimisation, while the GA has more commonly dealt with problems where real-valued parameters are used;
- In using these methods, there is neither need to know anything about the problem to be solved nor to assume all quantities are exact or deterministic;
- There is considerable scope for parallelisation of these methods, and in the case of GAs in particular such concepts have let to interesting new ideas of 'spatial' as well as 'temporal' population interaction;
- It is nonetheless true that in many cases the performance of the techniques can be considerably enhanced by building in some problem specific knowledge;
- Like SA and TS, GA provides the user with several parameters to tune;
- In TS as in the case of SA, the heuristic does not know the optimum has been found and carries on exploring;
- In TS unlike SA, the search may not remain close to the optimum;

The above methods, illustrate that GAs differ substantially from more traditional search and optimisation methods. The most significant differences are [75]:

- GA searches a population of points in parallel, not a single point;

- GA does not require derivative information or other auxiliary knowledge. In fact, only the objective function and corresponding fitness levels influence the directions of search;
- GA uses probabilistic transition rules, not deterministic ones;
- GA can provide a number of potential solutions to a given problem;

The power of GAs come from the fact that the technique is robust, and can successfully deal with a wide range of problem areas, including those, which are difficult for other methods to solve. GAs are not guaranteed to find the global optimum solution to a problem, but they are generally good at finding "acceptably good" solutions to difficult problems "acceptably quickly". For this reason GA was selected as the optimisations tool for this vehicle routing and scheduling system. Having mentioned all that, it is nonetheless true that in many cases the performance of these techniques can be considerably enhanced. Appendix B, provides some recent developments into building more effective techniques:

3.5 Genetic algorithms and Vehicle Routing and Scheduling (VRSP) Formulations

As indicated earlier VRSP is classified as a NP-Hard combinatorial problem [65, 101]. It is very difficult to solve these combinatorial problems to optimality and their computational burden exponentially grows with the problem size. Accordingly, any GA application for combinatorial problems could be useful for addressing these problems. This section is devoted to the application of genetic algorithm in related combinatorial fields to this research such as transportation, vehicle delivery scheduling, vehicle scheduling, vehicle routing, and logistics problems. The aim is to investigate how the fitness function is formulated for GA operations and what criteria and constraints are considered in the respective formulation.

3.5.1 Genetic algorithms in Transportation Problems:

Vignaux and Michalewic (1991) [102] developed two systems known as Genetic-1 and Genetic-2 to address a linear transportation problem. In Genetic-1, they used classical genetic algorithm, where they used bit strings to encode chromosomes [list of 0's and 1's]. A bit vector was defined for a solution in the transportation problem. Vectors are presented as $[V_0, V_2,\ldots,V_p]$, $[P= n.k]$, such that each component V_i $[1=1,2,\ldots,p]$, is a bit vector

$[w_0^i, ..., w_s^i]$ and represents an integer associated with row j and column m in the allocation matrix, where $j = \lfloor (i-1)/k+1 \rfloor$ and $m = [i-1] \bmod k+1$. The fitness function was expressed as the total cost of transporting items from sources to destinations and it is given by the formula:

$$eval(\langle v_1, v_2, ..., v_p \rangle) = \sum_{i=1}^{p} v_i . \cos t[j][m] \qquad \text{.....................} (1)$$

where,

n = number of sources,

k = number of destinations,

$P = n.k$

i = sources,

j = destinations,

cost (i,j) = the unit transportation cost between source i and destination j

However, there is no natural definition of genetic operators for the transportation problem with the above representation. Application of any GA operators would result in violation of constraints and results in loss of symmetry in expressing the constraints.

The Genetic-2 approach was based on matrix structural representations. In this approach the fitness function was expressed as:

$$eval(v_{ij}) = \sum_{j=1}^{k} \sum_{j=1}^{n} v_{ij} . \cos t[j][m] \qquad \text{...............} (2)$$

This formulation was easier than that in the Genetic-1 approach. They concluded that the matrix-based algorithm performed better than the vector-based version.

<u>**Cadenas and Jimenez [1994]**</u> [66] proposed a Genetic algorithm based solution method to the case in which fuzzy goals were assumed in the multi objective solid transportation problem. Fuzzy sets were used to model data that are not known with certainty by the decision makers. They formulated the fitness function as the objective function of the linear programming problem:

$$eval(v_{ijk}) = \sum_{p=1}^{p} w_p \mu_p (Z_p) \qquad \text{... ....} (3)$$

$$0 \le w_p \le 1, \text{ for } p = 1, 2, ..., P$$

$$\sum_{p=1}^{p} w_p = 1 \qquad \text{..............} (4)$$

Where,

$i = 1,2, ..., m$ (sources),

$j= 1, 2, \ldots, n$ (destinations),

$k=$ means of conveyance

$P =$ decision criterion

Also, $\mu_p(Z_p)$ is a membership function corresponding to p-th objective. Also, w_p represents the weight factor to reflect the relative importance of each objective.

Jimenez and Verdegay (1996) [67] proposed a Non-standard Genetic algorithms (NGA) based solution method to a multiobjective solid transportation problem, considering possible non-linearity in the objective functions. In this approach they suggested the use of intervals for uncertain data rather than typical point values for data such as supplies demands and conveyance. In their approach the objective function was formulated as:

$$Z = \sum_{p=1}^{p} w_p \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} f_{ijk}^{p}(x_{ijk}), \qquad \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

for p=1,…, P

Subject to:

$$\sum_{j=1}^{n}\sum_{k=1}^{K} X_{ijk} = A_i....for...i = 1,2,...,m \qquad \ldots\ldots\ldots (6)$$

$$\sum_{i=1}^{m}\sum_{k=1}^{K} X_{ijk} = B_i....for...j = 1,2,...,n \qquad \ldots\ldots\ldots\ldots (7)$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ijk} = E_k....for...k = 1,2,...,n \qquad \ldots\ldots\ldots\ldots (8)$$

$$X_{ijk} \geq o....for...all..i, j, k \qquad \ldots\ldots\ldots\ldots (9)$$

Where,

$i = 1,2, ..., m$ (sources),

$j= 1, 2, \ldots, n$ (destinations),

$k=$ means of conveyance

$P =$ decision criterion

$X_{ijk} =$ the amount of product transported from source I to destination j by means of conveyance k

Where p represents the *p-th* decision criterion, $A_i = [a_i^1, a_i^2]$, $\mathbf{B}j = [b_i^1, b_i^2]$, $E_i = [e_k^1, e_k^2]$, for all *i, j, k* are intervals of possible (i.e. admissible) values for the supplies, demands and conveyance capacities respectively. They also used the weight approach to reflect the relative importance of the objectives. They claim that this approach does not need more additional memory than the one used for transportation problems with point values. Also they claim that this method finds better solutions for non-linear problems.

3.5.2 Genetic algorithms in Vehicle delivery Problems:

Elmahi et al. (2004) [103], proposed a genetic algorithm that provides the optimal or nearest optimal, sequence of shipments within a supply chain under the just in time setting. In this approach the due goods have to be transported to final customers according to a predefined set of due dates while considering the loading capacities of transporters. In this approach the shipments are performed in batches and this is considered as batches delivery scheduling for a single supplier single customer. To evaluate a sequence of shipments a function was introduced to compare the real arrival dates of the requested products to the due ones. The evaluation focusess mainly on how many solutions can respect the just in time criteria and therefore it measures the advanced time of deliveries.

$$A\,[P_{j,\,i,\,k}] = y_d\,[P_{j,\,i,\,k}] - y_r\,[P_{j,\,i,\,k}] \qquad \text{.........}\ (10)$$

The term $A\,[P_{j,\,i,\,k}]$ indicates the amount of time with which, the k^{th} product (P) of the i^{th} batch in the j^{th} supply link, arrives before its due date. Y_d, represents the specified products due date time and y_r, indicates the real arrival time for the product respectively. The objective function was defined to minimise the total advanced time for the whole demand as follows:

$$A_g = \sum_{p=1}^{D} A(P) \qquad \text{.........}\ (11)$$

Jih et al. (1999), [104] hybridised GAs with dynamic programming in order to take advantage of the properties of each approach in solving the dynamic single-vehicle pickup and delivery problem. This approach considered both time window and capacity constraints. The dynamic programming was used to generate the optimal routes. If optimal solutions are not found within the specified time slot, the partially constructed routes are transferred to the GAs. These routes provide the basis for generating an initial population that often leads to better convergence than a randomly generated initial population. The presented approach enables dynamic programming to achieve real time performance and GA to approximate optimal solutions. In this approach the objective function was defined as:

$$Z_{dynamic} = Z_{Objective} + \left[\alpha_1 \sum_{r \in V^+ \cup V^-} f_{delay}(r) + \alpha_2 \sum_{r \in V^+ \cup V^-} f_{overload}(r) \right] \quad \text{.........} (12)$$

$$Z_{Objective} = \omega_1 \sum_{r,s \in V} d_{rs} x_{rs} + \omega_2 \sum_{r \in V^+ \cup V^-} f_{waiting}(r) \quad \text{.........} (13)$$

$$x_{rs} = \begin{cases} 1, & \text{if arc(r,s) is used by vehicle;} \\ 0, & \text{Otherwise.} \end{cases} \quad \text{.........} (14)$$

The two factors w_1 and w_2 are the weights, which reflect the relative importance of the total travel time and total waiting time objectives. Also, α_1 and α_2 are penalty coefficients. The function $f_{wating}[r]$ defines the waiting time of loading and unloading locations [i.e. $r \in V^+ \cup V^-$] as:

$$f_{waiting}(r) = \begin{cases} a_r - t_r, & \text{if vehicle arrives at location } r \text{ early;} \\ 0, & \text{Otherwise.} \end{cases} \quad \text{.........} (15)$$

The function $f_{Dealy}[r]$ and $f_{Overload}[r]$ are defined as following:

$$f_{delay}(r) = \begin{cases} t_r - b_r, & \text{if a vehicle arrives at location } r \text{ late;} \\ 0, & \text{Otherwise.} \end{cases} \quad \text{.........} (16)$$

$$f_{Overload}(r) = \begin{cases} l_r - Q, & \text{If the current load } l_r \text{ exceeds the vehicle capacity Q;} \\ 0, & \text{Otherwise} \end{cases} \quad \text{.........} (17)$$

Garcia et al. (2002) [105], developed a genetic algorithm to address a scheduling problem involving a set of orders by a homogenous vehicle fleet under the assumption that orders were required to be manufactured and delivered immediately to the customer site within a given time window. It is assumed that each vehicle delivers no more than one order at a time. Also, the order size is smaller than the vehicle capacity.

In this approach the objective function is considered to maximise the value of orders that are selected to be served within their time windows. When an order is not served at its ideal due date, a decrease of the orders original value proportional to its deviation is considered. Assuming W_i as the profit associated with serving order i at instant e_i [e_i = product due date], then W_i^- and W_i^+ are used to represent the earliness and tardiness penalties, which are used to decrease the profit or value when order i is served prior to, or after the ideal due date [i.e. S_i], respectively. Thus, when an order is served at instant $S_i + r$, the profit of order

i becomes $W_i - [r\text{-}S_i]\, W_i^+$. During each iteration, the chromosomes are evaluated and given a fitness value based on their respective profit.

3.5.3 Genetic algorithms in Vehicle Scheduling Problems:

Malborg (1996) [106], investigated the potential advantages of the application of a GA to a service level based vehicle scheduling problem. The objective was to minimise the unit periods of waiting (UPW) between the accumulation and delivery of correspondence between work centres. The objective function considered for this problem was complex and it was based on a number of terms as follows:

1. The expected volume of correspondence for destination j that is retrieved on the k stop at destination i can be calculated using:

$$\sum_{i=1}^{n}\sum_{k=1}^{d_i-1}\sum_{j=1}^{n}\left(\sum_{t=d_{i,k-1}}^{d_{ik}} v_{it}\, p_{ij}\right) \quad \text{.........} (17)$$

n = the number of destinations served by the vehicle.

T= the length of the operating period in time units.

V_{it} = the expected volume of correspondence accumulating at work centre i during time period t, for $i = 1, \ldots, N$ and $t=1,\ldots,T$.

P_{ij} = the probability that a unit of correspondence accumulating at work centre i is destined for work centre j, for $i, j=1,\ldots,n$.

n = the number of destinations served by the vehicle.

k = the number of departures from destination i.

d_{ik} = the rank ordered time of the k-th stop at destination i, for $i=1,\ldots,n$ and $k= 1,\ldots,k_i$

2. The expected UPW for the correspondence prior to pick up at destination i can be estimated using:

$$\sum_{t=d_{i,k-1}}^{d_{ik}} v_{it}\, p_{ij}\,(d_{ik} - t) \quad \text{.........} (18)$$

3. The travel time before this correspondence is delivered to destination j can be obtained by the difference between d_{ik} and the first departure time from destination j following d_{ik}:

$$\min_{m}\{d_{jm} - d_{ik} > 0\} \quad \text{.........} (19)$$

In this formulation, it was assumed that all the correspondence accumulating during the operating period must be picked up and delivered. Therefore, it was necessary to include a final pick up and delivery at each destination after the end of the operating period.

Malmborg (1996) [106] used the above definitions in an objective function to minimise the total UPW associated with scheduling the vehicle as follows:

$$\text{Minimise} f(X) = \sum_{i=1}^{n}\sum_{k=1}^{d_i-1}\sum_{j=1}^{n}\left(\sum_{t=d_{i,k-1}}^{d_{ik}} v_{it}p_{ij}\right)\min_{m}\{d_{jm}-d_{ik}>0\} + \sum_{t=d_{i,k-1}}^{d_{ik}} v_{it}p_{ij}(d_{ik}-t) + \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\sum_{t=d_{i,ki-2}}^{d_{i,ki-1}} v_{it}p_{ij}\right)(d_{j,kj}-d_{i,ki-1}) \quad \text{............ (20)}$$

Subject to:

$$d_{i,ki-1} \geq T, fori = 1,...,n, \quad \text{............ (21)}$$

$$d_{i,ki-1} < d_{j,kj} for - i \neq j - andi, j = 1,...,n. \quad \text{............ (22)}$$

Park et al. (2001) [107] described vehicle scheduling problems with service due times and time deadlines where three conflicting objectives were considered. The fitness is calculated based on three objective functions. The weighted sum of these objectives is the combination of the normalised total travel time, the normalised total weighted tardiness, and the normalised fleet size. The weighted sum of three objectives is computed by:

$$f(S_k) = (w_1\frac{t_k}{t^o_{\max}} + w_2\frac{d_k}{d^o_{\max}} + w_3\frac{v_k}{v^o_{\max}})100, \quad \sum_{i=1}^{3} w_i = 1 \ \ \& \ \ w_i \geq 0, \ \ i = 1,2,3 \quad \text{............... (22)}$$

Where w_1 is the weight of the minimisation of the total travel time, w_2 the weight of the minimisation of total weighted tardiness, w_3 the weight of the minimisation of the fleet size, $t^o_{\max}$ the maximum total travel time, $d^o_{\max}$ the maximum total weighted tardiness among the initial chromosomes, $v^o_{\max}$ the maximum fleet size among initial chromosomes, t_k the total travel time of chromosome k, d_k the total weighted tardiness of chromosome k, and v_k the fleet size if chromosome k.

3.5.4 Genetic algorithms in Vehicle Routing Problems:

Potter et al. (1995) [108] proposed a two-level genetic algorithm to address vehicle routings in General Pickup and Delivery Problem (GPDP). In this work a public transport system was considered as a special case of the GPDP, where a fleet of vehicles is required to satisfy a set of transportation requests from customers in an optimum manner. This paper explores the use of genetic algorithms to find a near optimum solution as an alternative to dynamic programming and other algorithms in literature.

In this two-level approach, an upper level GA was used to allocate passengers to vehicles and a lower level GA was used to find the shortest route for a given set of passengers in a single vehicle. The objective function used to evaluate the fitness of chromosome is based on a linear combination of the vector of *waiting time* [*w*] and *travel time* [i.e. distance], [*d*] for all passengers [i.e. tasks] on the route. Therefore, the objective function for the lower level GA is given as:

$$f(w,d) = \min\left[\sum_{i=1}^{n}(C_1 w_i + C_2 d_i)\right] \qquad \text{.........} (23)$$

The upper level GA fitness function is the sum of the fitness values for each single vehicle used in the problem. C_1 and C_2 are the weighting factors used to reflect the relative importance of each of the considering parameters.

Ochi et al. (1998) [109] presented a new hybrid metaheuristic, which used Parallel Genetic algorithms and Scatter Searches coupled with a decomposition-into-petal procedure for solving a class of vehicle routing and scheduling problem. The performance of this approach was evaluated for a heterogeneous fleet problem rather than homogenous vehicle routing problem.

The heterogeneous problem considers a set $\psi = \{1,\ldots,k\}$ of different vehicle types. A vehicle of type k has a carrying capacity Q_k. Also the number of vehicles of type k is n_k, which is not limited $(n_k = \infty, k \in \psi)$. The cost of the travel from customer i to j $[i, j = 0,\ldots, n]$ with a vehicle type k is d_{ijk}, which is considered to be the same for all the vehicle types.

The use of one vehicle of type k implies a fixed cost f_k and different vehicle types will reflect different fixed costs.

The fitness function is based on the summation of the fixed costs and travel costs for the given fleet in a chromosome. The main objective is to determine a fleet of vehicles with the considering minimum cost. The objective function is given by:

$$[Q_i - D_i] \times f_i \qquad \text{......... (24)}$$

The algorithm analysis the possibilities given by the k types of vehicles and it chooses the type of vehicle t_i which presents the lowest value for the objective function, where D_i is the accumulated demand of the routes taken by vehicle t_i. [109] suggested, this approach presented some advantages when compared to the other algorithms developed in this field.

Baker et al. (2003), [110] presented a hybrid of a GA with a neighbourhood search method to address a basic Vehicle routing problem. This problem considers customers of known demands to be supplied from a single depot. The main constraints considered in this approach are related to vehicle capacity and limits on travelling distances. Unlike, some representations, this approach does not specify explicitly the exact route that each vehicle should follow. In fact, individual routes are implicitly specified by assigning customers (n) to vehicles (m). The chromosome for an individual solution has the form of a string of length n with each gene value in the range $[1, m]$. The fitness of a chromosome is based on the capacity (R_c) and distance constraints violations (R_d) and the objective is to minimise this violation. The convergence of the GA is accelerated by incorporating neighbourhood search strategies, resulting in significant improvements. Baker et al. [110] further suggested that it is better to view the GA more as means of diversifying the exploration of the solution space alongside the neighbourhood search.

3.5.5 Genetic algorithms in Logistics Problems:

Wen et al. (2002) [111] used GA to address logistics scheduling problems. They developed a schedule for delivering cargo items from one point (called a Base Points) to several different points (called Supply Points) by some transportation assets such as trucks. Every cargo item must be delivered to its destination within a predefined time window. Otherwise, there is a penalty or a failure to deliver it. In this approach chromosomes are used to represent the cargo item delivery sequence. The fitness of a chromosome is obtained by computing the amount of cargo (number of items) that can be delivered by assets according to the sequence of this chromosome. To evaluate the fitness, two fitness metrics were used:

- **Requests**: the request is accomplished only if all the cargo is delivered successfully.
- **Missions**: one mission can include many requests and a mission is accomplished if all the requests are fulfilled. Therefore, the problem requires that a large number of missions be finished.

In this approach the larger the fitness function value is, the better the individual. They [111] found this GA approach more effective when compared with the greedy algorithm.

3.5.6 Discussion:

Reviewing the above genetic fitness formulations for several VRSPs, a number of points were observed. As regards the transportation problems, these problems are mainly considered as minimisation of transportation costs. In these formulations there are no considerations for routing and scheduling. In fact, the main objective is to find the least cost assignment of items from supply to demand locations. The latter Multi-objective problem formulation provided a better approach in modelling transportation problems. However, in these optimisation problems there were no considerations given to decision criteria such as delivery times, quantity of goods delivered, inventory levels, unfulfilled demands, under used capacities and etc. addressing such objectives further assists decision making on a transportation plan.

Observing the delivery problem formulations, one could see the emphasis is mainly on formulating the temporal aspects of VRSP. These problems are mainly considering a set of tasks or orders to be scheduled. These problems are formulated as single objective linear optimisation problems. The boundaries are mainly set at travel time, waiting time and loading capacities. The above formulations provide a real way of modelling the temporal aspect of VRSP, but they fail to consider aspects such as route selection, service level, resource utilisation, multi vehicle and etc.

In addition, vehicle-scheduling problems are mainly formulated like delivery problems. The main objectives considered here are minimising travel time, waiting time at loading and unloading locations, late deliveries and also fleet size. It is evident that the formulation of fitness functions for GA operations is relatively complex. Also, the problem formulation is based on linear combinations of different objectives.

Furthermore, in vehicle routing problems, the considered formulations suggest the typical minimisation of criteria such as travel time, waiting time, travelled distance and also fixed cost, which reflects the possible use of trucks with different capacities. It was evident from these formulations that the objective functions are linear combinations of criteria. Also, the boundaries are left at distance, travel time, and vehicle capacity.

Finally, in logistics management, the fitness formulation was modelled based on maximising the number of deliveries by adhering to the set time windows. This formulation was a linear bi-criteria optimisation problem. It certainly lacks route selection, and resource utilisation considerations.

The above points suggest that in modelling a VRSP, the objectives were either based on distance or travel time minimisations. In general one can observe the lack of considerations for route selections, resource utilisation, unfulfilled demands, under used capacities, reliability of deliveries, fleet size, operational cost and human fitness. Also, multi-criteria considerations are limited to linear behaviours. Furthermore, the constraint considerations are mainly limited to supply sources, demand centres and conveyance capacities with linear relationships.

In this work, the fitness function formulation considers several contrasting criteria, consisting of different practical aspects of logistics management based on transportation, routing, scheduling and operational objectives. Also, the considering constraints includes limits on supply, demand and other operational constraints that exists in the real system. Further descriptions of this objective function are described in chapter 6.

3.6 Multi-Objective Genetic Algorithm:

The GAs for multi-objective optimisation problems are generally similar to a single genetic algorithm in every way, except from the fitness evaluation and selection mechanism.
[133, 168] proposed the Vector Evaluated Genetic algorithm (VEGA) for finding solutions of multi-objective optimisation problems. In the VEGA, a population is divided into disjoint subpopulations that are governed by different objective functions. In this method the selection procedure was performed independently for each objective, but crossover was performed across subpopulation boundaries. Although this scheme was simple to implement selection of individuals in each criterion tend to lead VEGA to find extreme solutions where only one criterion is optimised.

To overcome this problem of VEGA, [74] suggested the use of the Pareto optimality ranking method. This approach is known as Pareto GA. In this approach the population is ranked on the basis of non-domination. All non-dominated individuals in the current population are identified. These are placed at the top of the list and a highest rank is assigned to them. These points are then removed from contention and the next set of non-dominated individuals is identified and assigned the second highest rank. This process continues until the entire population is ranked. Thereafter the selection occurs according to the individual ranks. Also, Goldberg suggested the use of niche formation to maintain appropriate diversity to prevent premature convergence.

To this effect [113] proposed the Niche Pareto Genetic Algorithm by incorporating the concept of Pareto domination in the selection procedure and applying niching pressure to spread the population out along the Pareto front. In this mechanism, two candidates for selection are picked at random from the population. A comparison set of individuals is also picked randomly from the population. Each of the candidates is then compared against each individual in the comparison set. If one candidate is dominated by the comparison set and the other is not, then the latter is selected for reproduction. If neither or both are dominated by the comparison set, the fitness sharing method is used to choose the winner. The goal of fitness sharing is to distribute the population over a number of different peaks in the search space, with each peak receiving a fraction of the population on proportion to the height of that peak.

3.6.1 Mathematical representation of non-inferiority

Suppose there are two objective functions, $Z(X) = [Z_1(X), Z_2(X)]$, and two solutions X_1 and X_2 are to be compared. That is the two vector quantities $Z(X_1)$ and $Z(X_2)$ must be compared and if $Z(X_1) > Z(X2)$, then X_1 is better than X_2. Since $Z(X_1) = [Z_1(X_1), Z_2(X_1)]$ and $Z(X_2) = [Z_1(X_2), Z_2(X_2)]$ is better than X_2 if and only if:

$$Z_1(X_1) > Z_2(X_2) \text{ and } Z_2(X_1) \geq Z_2(X_2) \qquad \text{......... (25)}$$

Or

$$Z_1(X_1) \geq Z_1(X_2) \text{ and } Z_2(X_1) \geq Z_2(X_2) \qquad \text{......... (26)}$$

Suppose $Z_1(X_1) > Z_1(X_2)$ and $Z_2(X_1) < Z_2(X_2)$ then nothing can be said about the two solutions X_1 and X_2 and they are incomparable. The inferior solutions, which are dominated by at least one feasible solution, may be dropped. Non-inferior solutions are the alternatives of interest. A solution X is non-inferior if there exists no feasible y that $Z(y) \geq Z(x)$. If such a feasible solution $\boldsymbol{y}$ exists, then $\boldsymbol{x}$ is inferior.

3.6.2 Pareto-Optimal Genetic Algorithm:

In this application, the Pareto-optimal genetic algorithms was designed aiming to search for a set of non-dominated solutions in a decision space comprised of three main decision variables namely trucks, routes and start time. The objective function (i.e. vector) used in this application is composed nine components. The algorithm is demonstrated in Figure 3-20 and it works as follows. Also to support this algorithm, a hypothetical example on Pareto GA is provided in Appendix B.

1. **Initialization:**
   - An initial population of size ***n*** Schedules are created at random from the decision space.
2. **Dominance Check:**

- The current population is checked to see which individuals are non-dominated points.
- Two non-dominated points are indifferent to each other.
- Individuals, which are non-dominated, are assigned a Boolean value of true.

3. **Adjustment**
   - Any individuals in the current population, which are dominated, are replaced with an offspring created from two different non-dominated parents.
   - The parents are obtained at random from the current population, with equal probability of being chosen.
   - If there is only one non-dominated solution [there will also be at least one], then the second parent is obtained at random from the decision space.
   - When each chromosome is selected then the reproduction process such as crossover and mutation can occur, with a probability of occurrence defined by the user.
4. **Dominance Recheck**
   - The adjusted population is rechecked, to see which solutions are non-dominated.
5. **Increase Population**
   - If all of the individuals of the current population are non-dominated, **n** new offspring are created and added to the population.
   - The offspring are created in the same manner as described in step 3 of the GA.
6. **Repeat**
   - The algorithm is repeated by going back to step 2.
   - The algorithm can be terminated by the number of generations or the number of dead runs specified by the user.

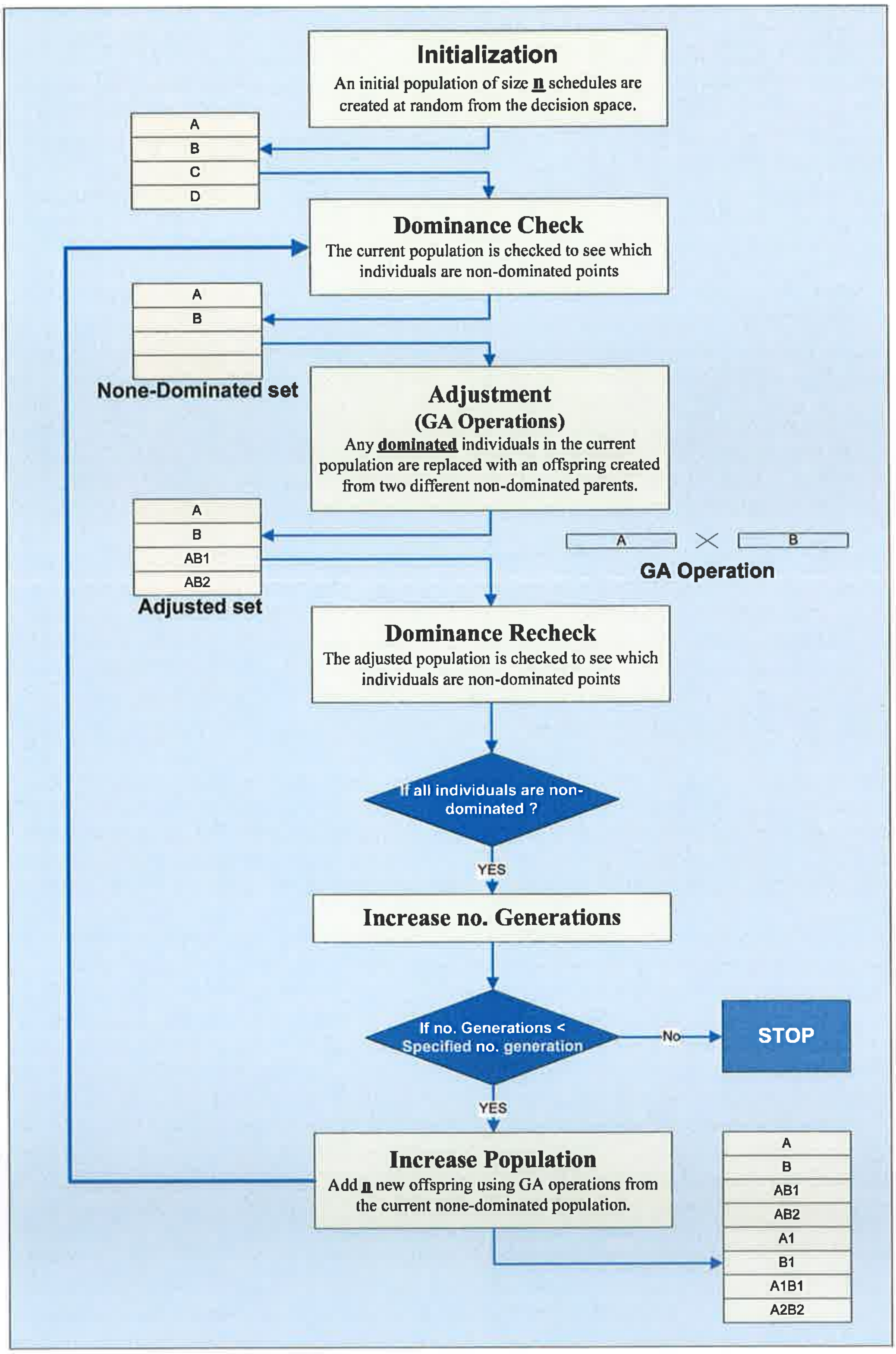


Figure 3-20 Pareto-Optimal genetic algorithm

3.7 Recent developments in genetic algorithms

Over the last decade considerable research has been conducted to improve the overall GA performance. Amongst others, one can refer to the advancements in selection, reproduction, encoding, adaptive and parallel GA techniques. For further information on these techniques the reader is asked to refer to Appendix B.

3.8 Conclusions

To address the shortfalls of conventional approaches to VRSP, this chapter proposed the development of an integrated DSS. The DSS is composed of a search engine and a simulation model from the considering supply chain. The search engine was designed based on the GA principles. The aim of this application is to provide a set of optimum or near optimum transportation schedules. The simulation engine provides a more flexible way to incorporate system behaviours in terms of real constraints, stochastic and dynamic behaviours. GA as the main component of the search engine provides potential benefits in addressing the specified shortfalls. This method could greatly help in addressing multi-modal VRS optimisation problems considering either linear or non-linear relationships within objectives and constraints of the system. Also, the GA tends to provide a pool of solution, which could be used as alternative approaches to the optimum solution.

Also, this chapter presented the main building block of GA. It described genetic representation, initial population, fitness function, reproduction, selection schemes, GA operators and control parameters. As part of this chapter, further investigation was carried out on the use of GA in VRSP. For this reason the application of GA in the field of transportation, vehicle delivery scheduling, routing and logistics was studied. This work presented how fitness function was formulated in each field and it specified the main criteria and constraints considered in each field. The existing problem formulation presented a number of drawbacks as follows:

- In these formulations there is a general lack of considerations for route selections, resource utilisation, and unfulfilled demands, under used capacities, reliability of deliveries, fleet size, operational cost, inventory levels and human fitness considerations.

- The multi-criteria considerations are limited to linear behaviours.
- The constraint considerations are mainly limited to supply sources, demand centres and conveyance capacities. Also, constrains were subjected to linear behaviour.

As part of this research further investigation were carried out in multi-objection optimisation problems. To address Multi-objective optimisation problems, three approaches were introduces such as VEGA, Pareto GA and Niche Pareto GA. To this effect the development of Pareto-GA algorithm for this problem were presented. Based on the presented GA principles, the next chapter further illustrates how these concepts were used to develop the Search engine of the proposed DSS.

Chapter 4: Genetic Search Engine Design and Development

4.1 Introduction

The main aim of this project was to design and develop a hybrid optimisation-simulation based decision support system to aid logistics management in vehicle routing and schedule problems. For this reason an application was developed and it was implemented in a LPG distribution company, which is mainly involved in transporting LPG from Gas refineries to its bottling plants for processing and distribution to end users. The search engine and the simulator are integrated and they are also designed to operate as stand alone tools. Using this tool one could define and describe a system in terms of its supply and demand capacities. Also, the user could define possible routes linking the sources to demand centres and further more to describe different considerations for route selection. Moreover, the user could search for possible near optimal transportation schedules considering different objectives and constraints in the system and finally, Evaluate alternative transportation schedules and measure their impact on systems performance measures.

The search engine was developed using the Microsoft Visual Basic 6 programming language and the simulation model was developed using the Witness simulation tool package. Witness presents a number of features, including integration capabilities and acting as an OLEII server making integration with Microsoft Access and visual basics achievable.

The intend of this chapter is to demonstrate how the search engine and its components including random search, genetic algorithm and the pareto optimal genetic algorithm were designed and developed for this application. This chapter first describes the architectural components of the search engine. Then each of these components is described briefly. The main search mechanism are presented. The main components of these methods are introduced, and it was shown, how these components were implemented. Also, different cost components considered to evaluate the fitness of transportation schedules are introduced. Furthermore, the objective function formulation was presented and finally the constraints and the penalty method to compensate for the violation of system constraints are presented.

4.2 Search Engine design components

This application was designed to work either as a stand-alone tool or in conjunction with a simulation tool. Therefore, this application consists of a number of different modules as shown in Figure 4-1. These modules are such as 1) The Search engine user Interface, 2) the search engine, 3) The Search engine- Database Interface, 4) The simulator user interface, 5) the Witness Simulation Module, 6) The Simulator- Database Interface.

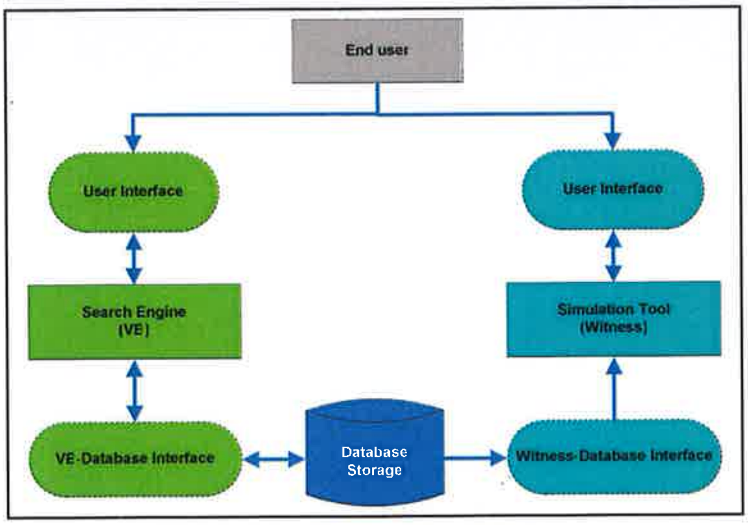


Figure 4-1, The Simulator Interfaces.

The following sections describe different components of the ***search engine user interface design***. In addition Appendix C describes the interfaces, designed and developed for linking Database to GA search and simulation module.

4.3 The Search Engine User interface Design

Figure 4-2, represents the main components of the user interface designed for the search engine. The user can access different functionalities of the search engine through this interface. The following subsections describe the main functionalities related to ***cost parameters*** and ***search engine methods***. The remaining menu options are described in Appendix C.

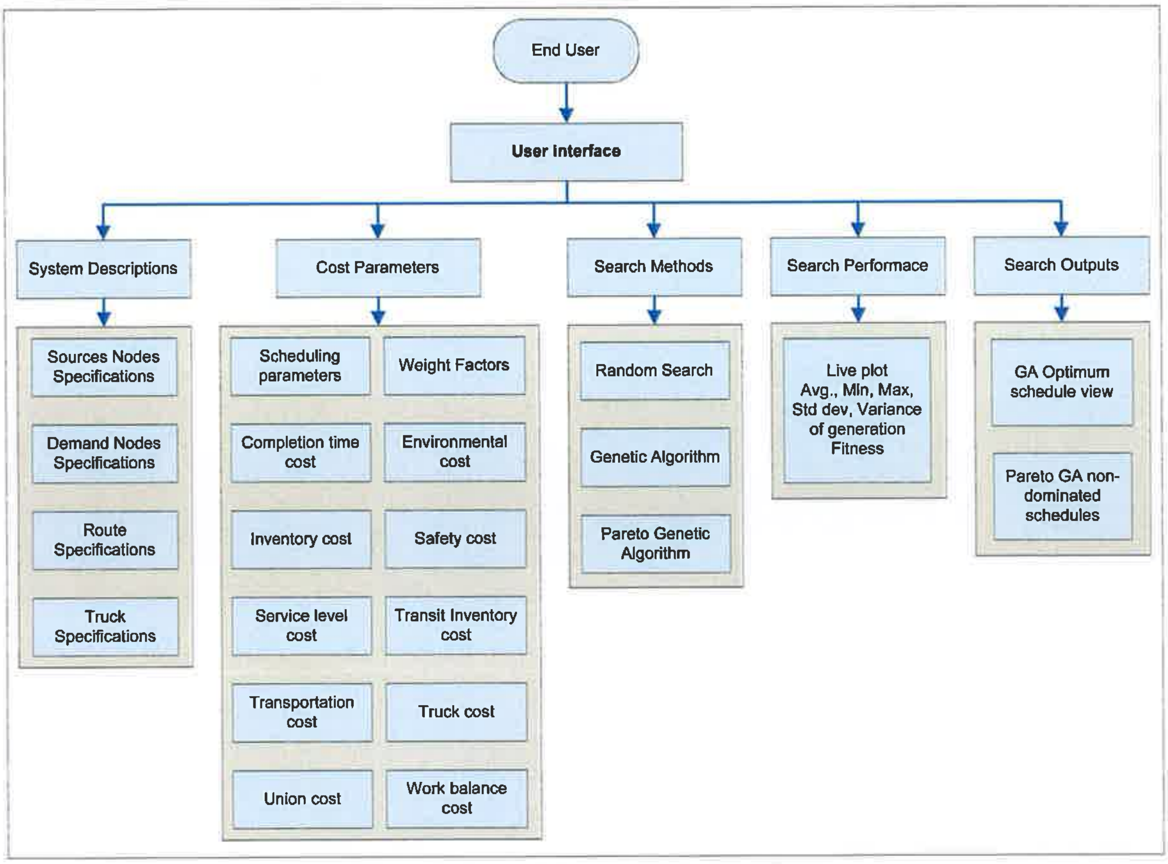


Figure 4-2 Search Engine User Interface

4.3.1 Cost Parameters

Once, a transportations system is defined using the ***System Description*** menu options. Then the user can use ***Cost Parameters*** menu option to define different objectives or criteria to be used for assessing the system. Basically, this menu option is used to define the objective functions for the transportation schedules. The defined objectives are used within GA search engine to evaluate the fitness of the generated schedules. For the considering system, there are nine main cost functions defined. These are such as completion time, environmental, inventory, safety, service level, transportation, truck, human factor and work balance cost measures. The user can add or remove any parameters to be considered in evaluating fitness of generated schedules. Figure 4-3, illustrates the menu options to define, add or remove a cost measures in the objective function.

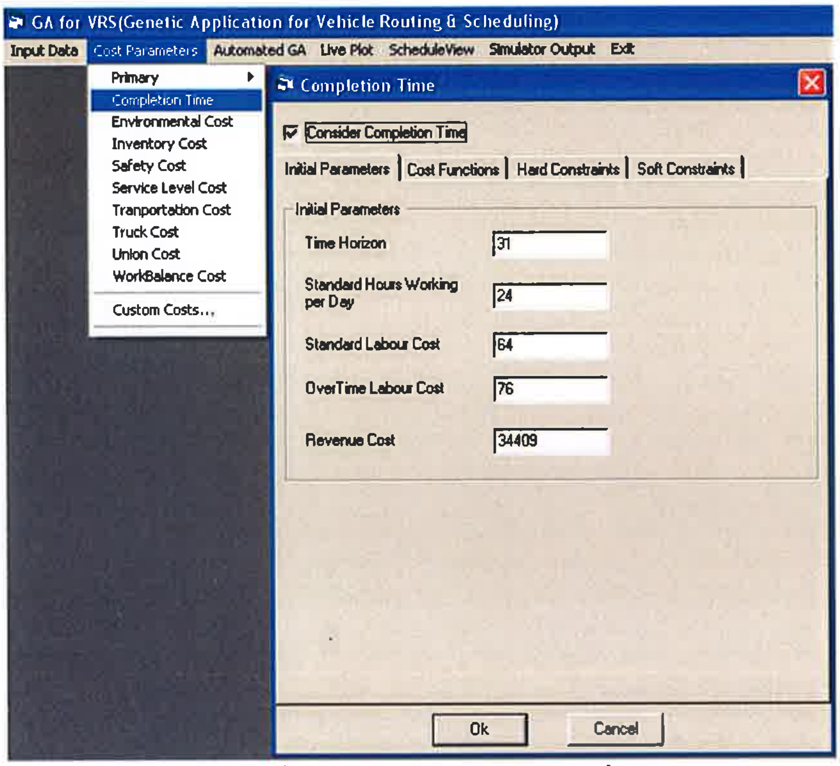


Figure 4-3 cost parameter menu options

Once, a cost measure is selected, the user is prompted with further menu options. These menu options allow one to further define specific cost measures. The main components of this user interface are such as: cost parameters, cost functions, hard and soft constraints and linear or non-linear based calculations. Each of these components are further described as follows:

4.3.1.1 Initial Parameters

This option allows the user to define a cost measure (i.e. an objective function) based on its basic input parameters. These metrics would be used in a function to calculate the value of the specified cost measure or objective. For instance, as indicated in Figure 4-3 the input parameters of completion time is based on time horizon, standard working hours per day, standard labour cost, overtime labour cost and revenue cost. The user has to determine these parameters before editing the cost function.

4.3.1.2 Cost Functions

This menu option was primarily designed to allow the user to edit and define an objective function based on its main parameters defined earlier. The user is prompted to define any formula to be used for the calculation of the specified cost measure. Figure 4-4 further illustrates this menu option.

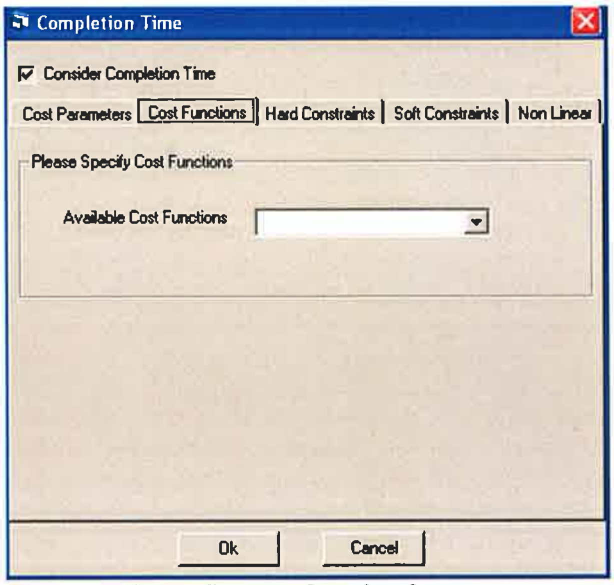


Figure 4-4 to define cost function for cost measure

4.3.1.3 Hard Constraints

Typically, logistics systems are subject to many real constraints. The aim of this menu option is to let the user define the possible existing constraints for the specified cost function. In general, constraints can be modelled as either hard or soft constraints. One can define any or both types of these constraints for the defined cost measure. Usually, violation of hard constraints results in infeasibility of the solution. However, violation of soft constraints results in encountering penalties and therefore addition of extra cost to the overall objective function.

Using this menu option one can typically set lower and upper bonds on the specified cost function. Furthermore, the user is prompted to edit any formula for restricting the cost object. Figure 4-5 demonstrates the menu option used to defined hard constraint for completion time. Section 4.7 and 4.8 provide more details on those constraints considered for the considering cost measures.

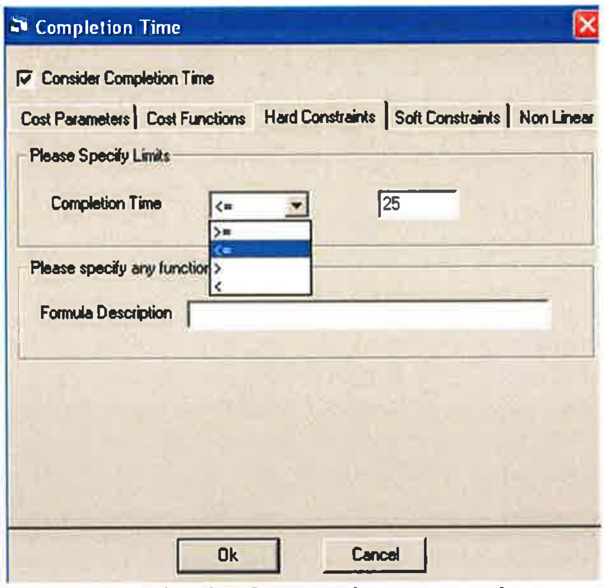


Figure 4-5 Hard constraint menu option

4.3.1.4 Soft Constraints

In addition to the hard constraint the user can define soft constraints for cost measures. Violation of such constraints results in adding extra penalties to the cost of the schedule. Constraint can be specified based on the lower and upper bonds defined. Penalties are usually defined in two methods.

__Method A:__ this method is based on defining a series of limits and the penalty associated with the violation of such limits. This method can be formulated as follows.

$$
\begin{aligned}
f(x) &= 0 && \text{if } 0 < X_{ij} \le 2 \\
&= C_{ij} && \text{if } 2 < X_{ij} \le 4 \\
&= 2\,C_{ij} && \text{if } 4 < X_{ij} \le 6 \\
&= 3\,C_{ij} && \text{if } 6 < X_{ij} \le 8 \\
&= 4\,C_{ij} && \text{if } 8 < X_{ij} \le 10 \\
&= 5\,C_{ij} && \text{if } 10 < X_{ij}
\end{aligned}
\qquad \ldots\ldots\ldots\ldots\ldots\ldots (27)
$$

Where,

C_{ij} = Cost and

X_{ij} is the actual value for the objective cost

Figure 4-6, illustrates how the menu options could be used to define Method A type of constraints. Also Figure 4-7 graphically demonstrates such limits.

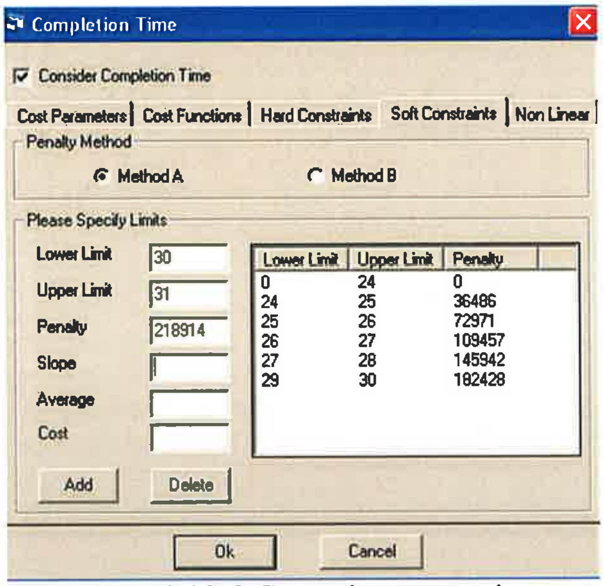


Figure 4-6 Soft Constraint menu option

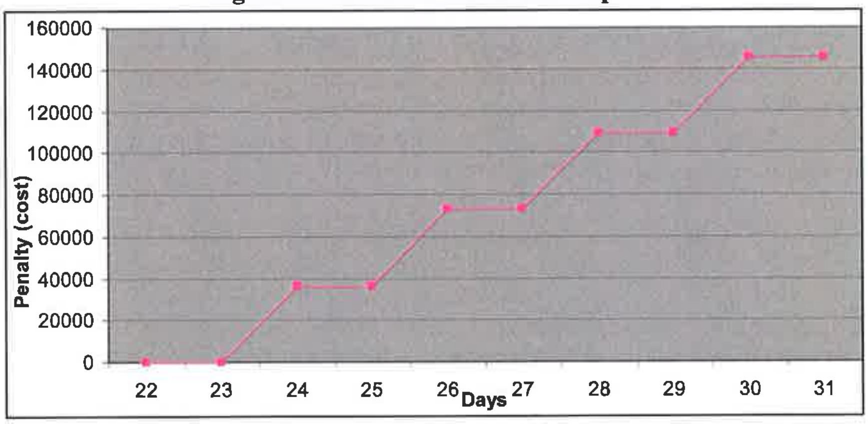


Figure 4-7 soft constraints defined for completion time cost measure

<u>Method B:</u> this method is very similar to the last method. Using the Slop, Average and Cost parameters in the menu option the user can formulate the penalty cost using the following formula.

$$f(x) = C_{ij}(1 + 0.2(X_{ij} - Average)) \qquad \text{..................} (28)$$

Where, C_{ij} = Cost factor, X_{ij} = Objective function value

The slope here is set to 0.2 and the average is also is the expected average value for the considering cost function. Further to these methods, the user can set non-linear penalties by applying any of the non-linear functions such as exponential, Sin and Natural Logarithm functions developed for this application.

4.3.1.5 Non-linear Objective function

The above soft constrain formulation allows defining non-linear penalties for the considering cost functions. Also, the non-linearity could be applied to the Objective function itself. In this approach, the user can choose any of the non-linear methods to map the obtained cost function to a non-linear cost measure. The non-linear options are developed based on the Exponential, Sin and Natural logarithm functions. Figure 4-8 shows how menu options can be used to handle non-linearity within the objective function.

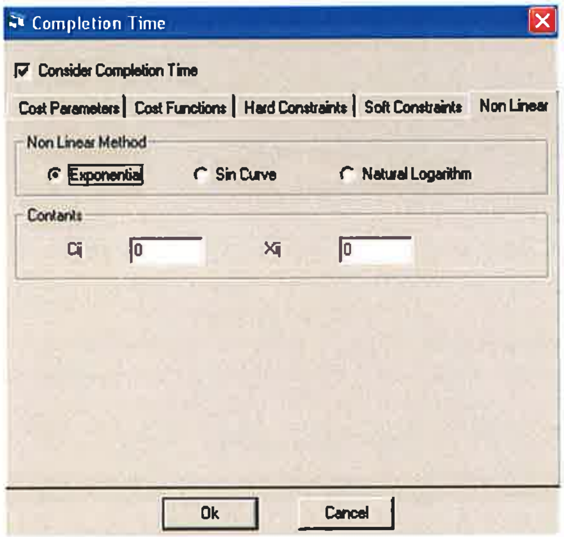


Figure 4-8 Non-linear functions

- **Exponential Curve:** This function could also be used as square root curve too. Figure 4-9 and Figure 4-10 illustrates these functions graphically. The general formulation for this function is as follows:

$$C_{ij}X_{ij}^{\alpha} \qquad \text{................ (29)}$$

C_{ij} = Cost value to be specified by the user.

X_{ij} = The calculated parameter such as distance, completion time, safety and etc.

α = Intensity level to be specified by the user [e.g. 0.5, 2, etc.]

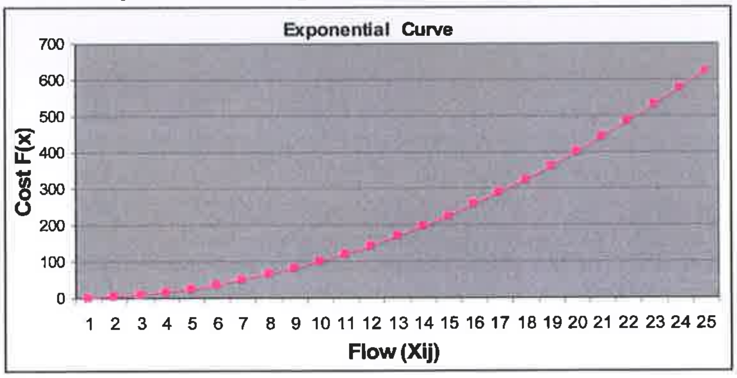


Figure 4-9 Exponential Function

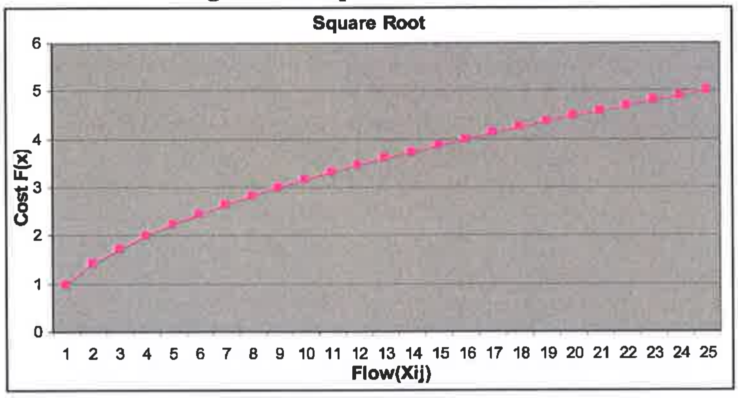


Figure 4-10 Square function

- **<u>Sin Curve:</u>** The general formulation for this function is as follows. Figure 4-11, illustrates this function graphically.

$$C_{ij}X_{ij}\left(Sin\left(X_{ij}\frac{\pi}{4}\right)+1\right) \qquad \text{......................} (30)$$

C_{ij} and X_{ij} are defined as in the above equation.

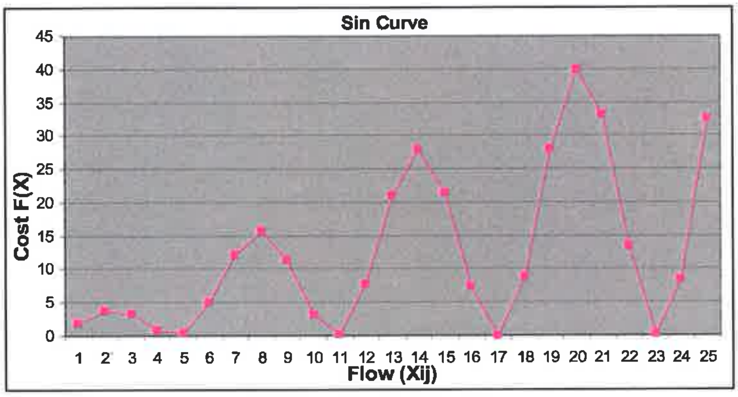


Figure 4-11 Sin Function

- **<u>Natural Logarithm Curve:</u>** This equation is based on logarithm function as follows. Figure 4-12 demonstrates this function graphically.

$$C_{ij} * (Ln(X_{ij})) \qquad \text{......................} (31)$$

C_{ij} and X_{ij} are defined as in the above equation.

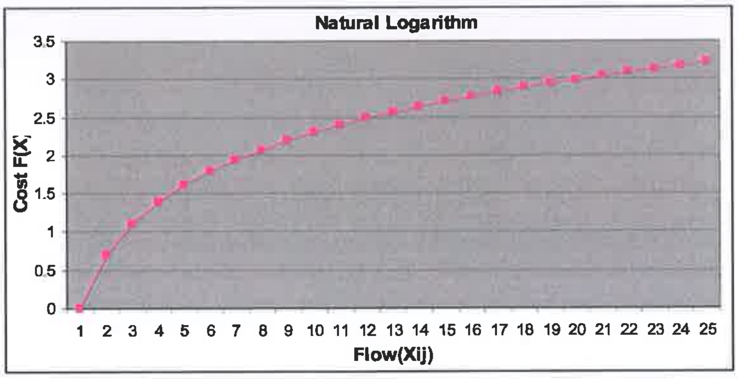


Figure 4-12 Natural Logarithm function

4.3.2 Search Engine Methods

This menu option allows the user to access any of the developed search engines here within this application. The search mechanisms are such as ***Random***, ***Genetic*** and ***Pareto Optimal Genetic*** searches. Figure 4-13, illustrates the menu option that provide the access to these search methods. Each of theses search methods is further described in the proceeding sections.

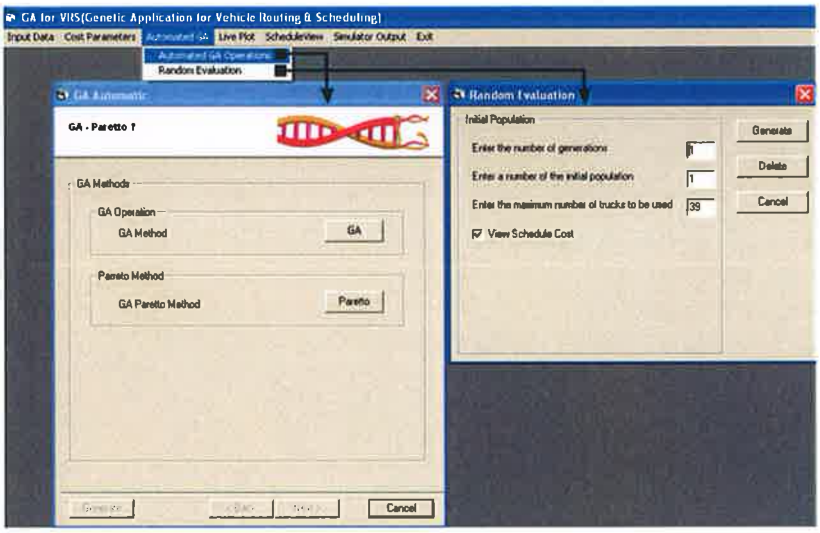


Figure 4-13 Search Engine Menu option

4.4 Random Search Engine

As described earlier in chapter 3, the random search method was developed to evaluate the performance of the Genetic search engine. As illustrated in Figure 4-14, the user needs to identify parameters such as: the total number of generation, the initial population size and the total number of trucks that could be used as the main resource in the schedule.

The steps involved in creating schedules based on the random method are exactly the same as creating initial population as in the genetic search engine. These steps are further described in section 4.5.3. In this approach, an initial population indicates the number of chromosomes (i.e. schedules) required to establish a generation. In this approach, a new set of initial population is randomly generated in every generation. The general statistics are recorded per generation. Also, the best-found schedule and the most and the least expensive schedules are recorded and updated per generations. This method randomly searches the solution space without having any particular approach to find the optimum cost schedule. Figure 4-14, illustrates how to set initial parameters for this random search mechanism.

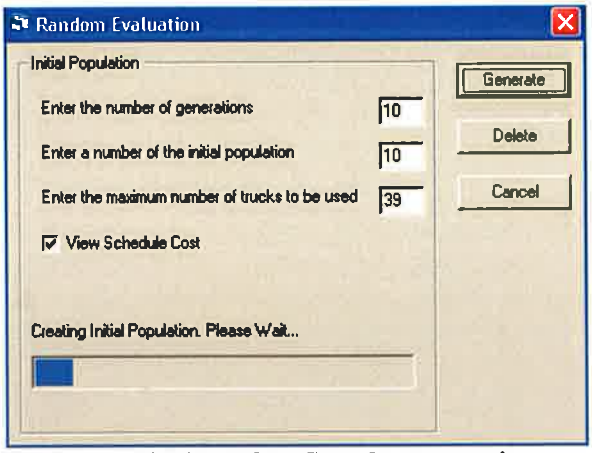


Figure 4-14 Random Search menu option

4.5 Genetic Search Engine

There are a number of variables to be specified before executing GA runs. These are population size (n) (*PopSize*) and generation number (t) (*GenSize*). Also, in this application, the user specifies the maximum number of resources available to use. Furthermore, to select individual chromosomes from a population, one must specify the selection method type, either roulette wheel or tournament selection method developed for this application.

To perform genetic operation, one must also specify, the genetic operators such as crossover and mutation types. To this effect, a number of global real variables including the probability of crossover (*Pc*), Probability of mutation (*Pm*) must be specified too. Finally, to evaluate the fitness of chromosomes (i.e. schedules), one must determine how to map the cost of the generated schedule to fitness values. Figure 4-15 illustrate a number of user menu options used to set initial parameters for GA search.

Figure 4-16, represents a single population composed of a number of chromosomes in a generation. Each chromosome has a real variable cost and a corresponding real fitness variable. In this GA application genetic operators are applied to an entire population at each generation as shown in Figure 4-17. Two non-overlapping populations are considered to implement these operators. This would simplify the creation of offspring and the replacement of parents. The new offspring are generated by applying the genetic operators on the old population (*OldPop*). The *OldPop* is entirely replaced with the new generated offspring and in this way the *OldPop* is set to new population (*NewPop*). Typically after each generation statistical report relating to the generation is provided. This report includes measures about the population such as average, maximum, minimum, standard deviation and variance of fitness of the population.

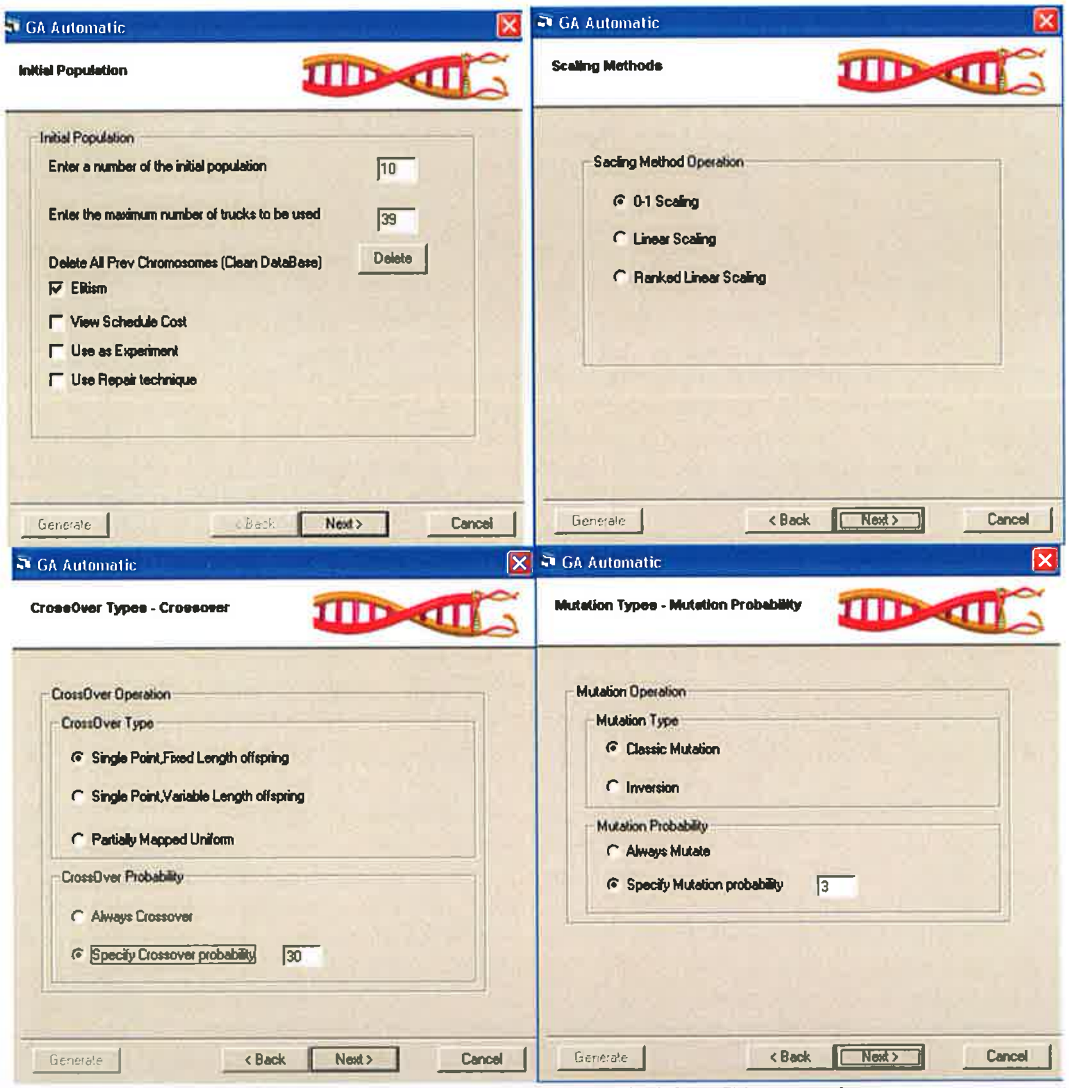


Figure 4-15 user menu options for initialising GA operations

1

Population			
Generation No.	Chromosome No.	Cost(x)	Fitness f(x)
1	1	15	6.67
1	2	9	11.11
1	3	8	12.50
1	4	7	14.29
1	5	20	5.00
1	6	30	3.33
1	7	40	2.50
1	8	50	2.00
1	9	16	6.25
1	10	80	1.25
1	11	12	8.33

1- Front view of a Population.

2- Breakdown of a chromosome.

3- Breakdown of a Route.

2

Chromosomes		
Chromosome No.	Trucks	Route No.
1	1	10
1	1	11
1	22	12
1	22	13
1	3	15
1	3	16
1	4	17
1	4	18
1	4	21
1	6	19
1	6	20

3

Routes			
Route No.	Start Location	EndLocation	Start time
10	A	B	3:00 PM
11	C	D	8:00 AM
12	A	D	6:00 PM
13	C	B	12:00 AM
15	E	F	10:00 AM
16	G	H	7:00 PM
17	A	F	10:00 AM
18	E	H	8:00 PM
21	G	F	12:00 AM
19	C	H	11:00 PM
20	E	B	11:00 AM

Figure 4-16 Schematic of a population in a generation in the GA

Gneration (T)		
Chromosome No.	Cost(x)	Fitness f(x)
1	15	6.67
2	9	11.11
3	8	12.50
4	7	14.29
5	20	5.00
6	30	3.33
7	40	2.50
8	50	2.00
9	16	6.25
10	80	1.25
11	12	8.33
Population Report		
Average		6.66
Min		1.25
max		14.29
Std		4.44
Var		19.71

Reproduction
Crossover
&
Mutation

Gneration (T+1)		
Chromosome No.	Cost(x)	Fitness f(x)
1	30	3.33
2	10.2	9.80
3	4.5	22.22
4	8.8	11.36
5	7.7	12.99
6	22	4.55
7	33	3.03
8	44	2.27
9	55	1.82
10	17.6	5.68
11	88	1.14
Population Report		
Average		7.11
Min		1.14
max		22.22
Std		6.44
Var		41.49

Figure 4-17 schematic of non-overlapping populations used in the GA

4.5.1 Chromosome Representation

The first and the most important step in GAs is how to encode chromosomes to represent solution to the considering problem. Here, a chromosome must represent a complete monthly schedule for transporting LPG from different refineries to processing plants in different localities. The main parameters considered in developing a monthly schedule for daily activities are:

- **Trucks** are the main resource for scheduling. There is an upper bond on the number of trucks that are available for transportation. The total number of trucks to be used in a schedule is obtained randomly.
- **Routes** are the possible links between supply sources and demand centres. Routes are randomly assigned to trucks.
- **Start Time** is the departure time that is assigned to trucks to leave a refinery or processing plant. The start times are usually chosen randomly from a time range between 1 to 24 hours.

Figure 4-18, illustrates a chromosome and its hierarchical breakdowns. The chromosome is a combination of genes and it is defined as an array of infinite dimensions.

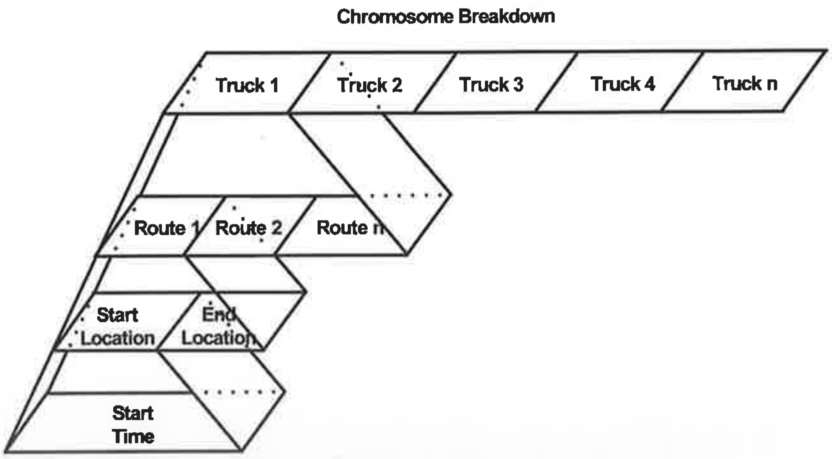


Figure 4-18 Hierarchical breakdown of a chromosome

The main decision variables are randomly chosen from their respective decision space to generate a schedule. Figure 4-19, represents a single schedule that is randomly developed for truck T1.

Trucks	Descriptions	CL	SL	EL	SL	EL
T1	Route Number		27	27	15	15
	Route Locations	Babol	Esfahan	Karaj	Abas	Qom
	Start Times (min)	420	1620	2820	3360	3920

Figure 4-19 a single transportation schedule for Truck T1

As shown in Figure 4-19, the routes are assigned to the Truck T1 randomly from the set of available routes. Each route includes a start location referring to supply points and an end location, which is the demand points or bottling plant in this case. The routes are identified using the route numbers. Also the start times are assigned to each location defined in the route. This suggests the departure time for the truck to the next destination defined along the path. Finally, the current location refers to the initial position of the truck before initiating the transportation schedule. Furthermore, Figure 4-20 illustrates a chromosome, which is a schedule for four trucks. As shown here the chromosome could have a variable length both in terms of number of routes assigned to a truck and also in terms of the number of trucks that are available in a schedule.

Trucks	Current Location	Start Location	End Location	Start Location	End Location	Start Location	End Location	Start Location	End Location
T1		27	27	15	15	19	19	28	28
	Babol	Esfahan	karaj	Abas	Qom	Arak	karaj	Esfahan	Kashan
	420	1620	2820	3360	3920	4480	5040	5600	
T2		12	12	14	14	11	11		
	Esfahan	Abas	Kashan	Abas	Mashhad	Abas	Karaj		
	60	520	1030	1540	2600	360			
T3		35	35	12	12				
	Karaj	Tehran	karaj	Abas	Kashan				
	180	340	1200	1740					
T4		23	23	18	18	22	22	37	37
	Kashan	Arak	Qom	Arak	EsfahanC	Arak	Mashhad	Tehran	Kerman
	1380	2440	3400	3475	4935	5995	6555	7115	

Figure 4-20 a chromosome (schedule) for 4 trucks

In this GA application, the genes of a chromosome such as trucks, routes and start times are directly chosen from the variable space. Therefore the proposed method of encoding is known as phenotypic. Also one may have used binary strings for encoding chromosomes. This approach is a genotypic, which requires further decoding of genes to obtain the solution in terms of variable space.

4.5.2 Fitness Evaluation

As described above, a schedule is a random assignment of the system's variables. Therefore, different combinations of the variables reflect different cost factors in the system. The cost parameters considered to evaluate the chromosomes' fitness are further described in section 4.7. Also, section 4.8 demonstrates how the fitness of a chromosome is formulated considering the cost parameters, systems constraints and weight factors. There is a need to map these cost measures to a fitness value. To map individual's cost to its fitness value, there are two options are provided as follows:

- **Mapping cost to profit:**

In this approach a chromosome cost is converted to profit using the following equation. Therefore, the fitness of a chromosome is directly proportional to its profit.

$$F_i = \frac{1}{\sum_{j=1}^{n} C_j}, i = 1, ..., m \ \& \ j = 1, ..., n \qquad \text{................ (32)}$$

Where C_j represents the cost measures considered within the objective function. The approach conducted here is to transform all the cost factors into one main single cost function. Then, the new formation of the problem would be to minimise the main cost function by satisfying all the existing constraints in the system.

- **Linear cost transformation method:**

In this approach, the fitness of each individual chromosome is calculated based on the most expensive chromosome that is present in the selection pool. This is calculated using the following equation.

$$\begin{aligned} f(x) &= C_{\max} - g(x) & \text{When } g(x) < C_{\max}, \\ &= 0 & \text{Otherwise,} \end{aligned} \qquad \text{.... ....(33)}$$

There are different ways to choose the coefficient $C_{\max}$ such as: It may be taken as an input coefficient, the largest ***g*** value observed thus far in the generations, the largest ***g*** value in the current population and finally the largest of the last ***k*** generations. In this application $C_{\max}$ is set to be the largest **g** value in the current population. Table 4-1 represents an illustration of this linear transformation. Figure 4-21, illustrates this fitness transformation clearly.

Table 4-1 Chromosomes costs and their respective fitness

Chromosomes	Costs	Fitness F(x) = Cmax - g (x)
1	178781	6758
2	132979	52560
3	158753	26786
4	79398	106141
5	84155	101384
6	78149	107390
7	49830	135709
8	73345	112194
9	185539	0
10	80323	105216

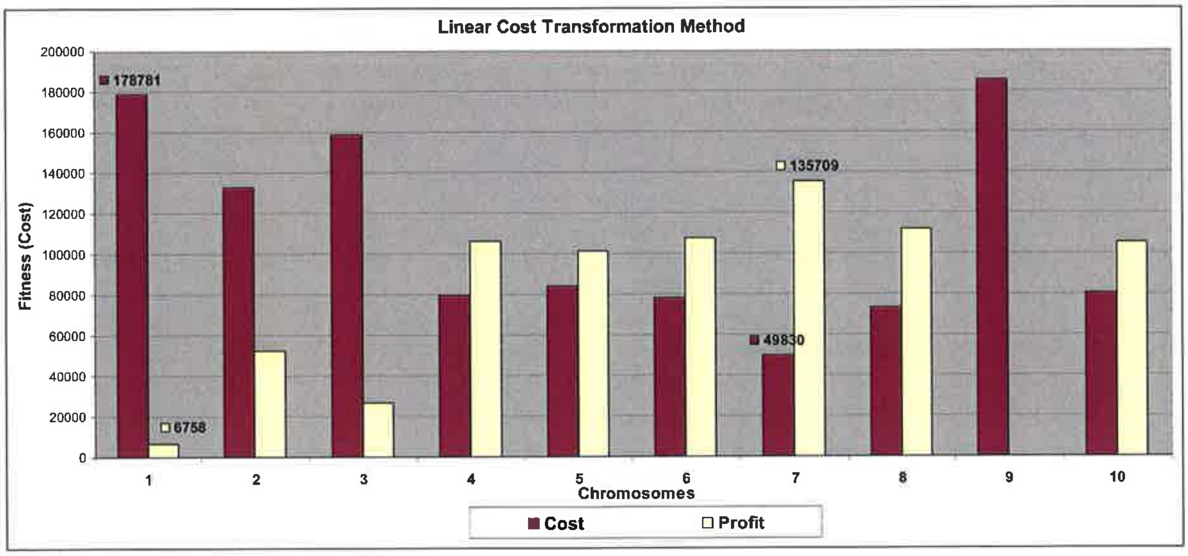


Figure 4-21 Impact of Linear cost transformations

Figure 4-22 illustrates steps taken to obtain the fitness of all the existing chromosomes within a population. Figure 4-23, represents a function developed to choose between the fitness transformation methods. Also, Figure 4-24 shows the function developed to map cost to fitness value.

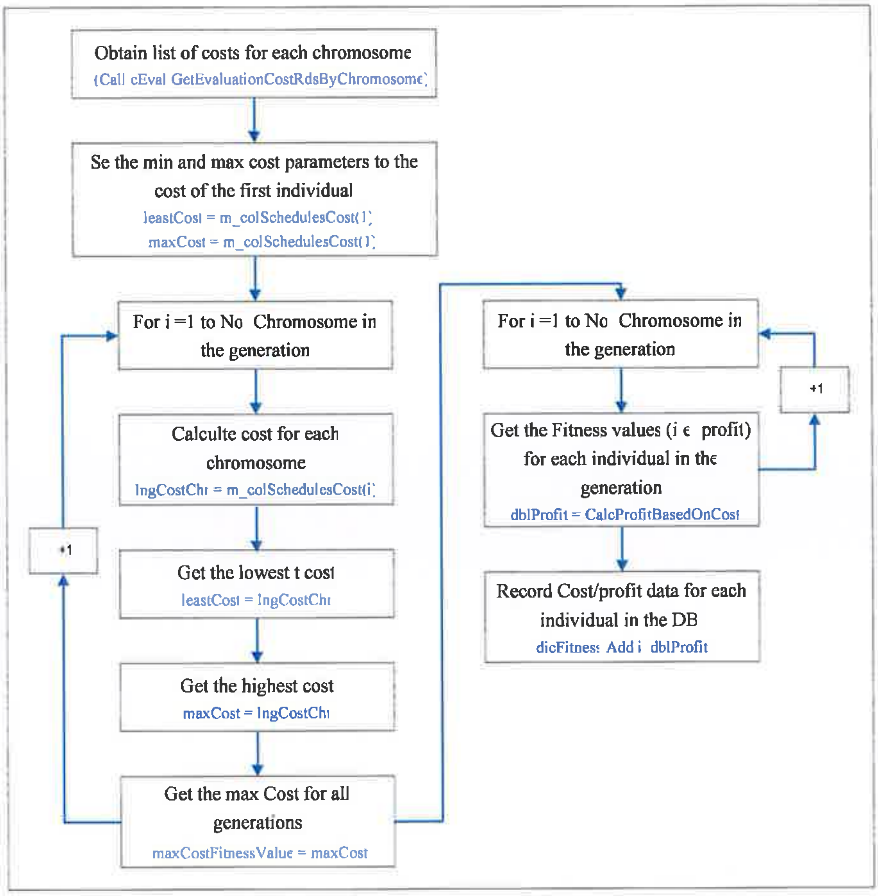


Figure 4-22 steps in chromosome fitness evaluation

```
Public Function CalcProfitBasedOnCost (costValue As Double, profitCalculation As
CostProfitMethodStyle, Optional maxFitnessVal As Double = 0) As Double
  If (profitCalculation = OneOverCost) Then
    CalcProfitBasedOnCost = ProfitReverseCost(costValue)
  Else
    CalcProfitBasedOnCost = Abs(maxFitnessVal - CDbl(costValue))
  End If
End Function
```

Figure 4-23, function to calculate schedule fitness

```
Public Function ProfitReverseCost(lCost As Double) As Double
  Dim iProfit As Double
  'calc profit
  iProfit = (1 / lCost) * (10 ^ 12)
  ' round up to 6 digits
  ' CostToProfit = Round(iProfit, 6)
  ' don't round up
  ProfitReverseCost = iProfit
End Function
```

Figure 4-24, function to convert cost to profit

The GA approach developed here purposefully allows the violation of soft constraints and therefore generation of infeasible solutions to extend the search space, through both feasible and infeasible regions for good quality solutions. The penalty method is used to relax the constraint of a problem and to solve an equivalent optimisation problem without constraints. The penalty calculations are described in section 4.8.

Also, it is important to note that violations of any defined hard constraint are not permitted. In this application, repair techniques are developed to restore chromosomes, violating any specified hard constraints. These techniques are further described in section 4.5.3.4.

During each generation, chromosomes are evaluated using the specified measure of fitness, the fitter a chromosome the higher its probability of being used in the genetic reproduction process. To compare a solution with the same fitness function the cost parameters and penalty factors are chosen before the beginning of the simulation and these are kept constant throughout the simulation. Thus from a given generation i to $i+1$, the fitness function which depends on these factors remains the same during the whole simulation.

4.5.3 Initial Population Generation

The initial population is a collection of chromosomes used to initiate the GA operations. In this application, to generate the initial population, the user must define the population size, referring to the total number of individuals (i.e. schedules) to be included in a population. Also, the user must specify the lower and upper bound on the number of transportation recourses (i.e. trucks) to be used in establishing chromosomes. As shown in Figure 4-25, there are six main steps taken to generate an initial population. These steps are further described in the subsequent sections.

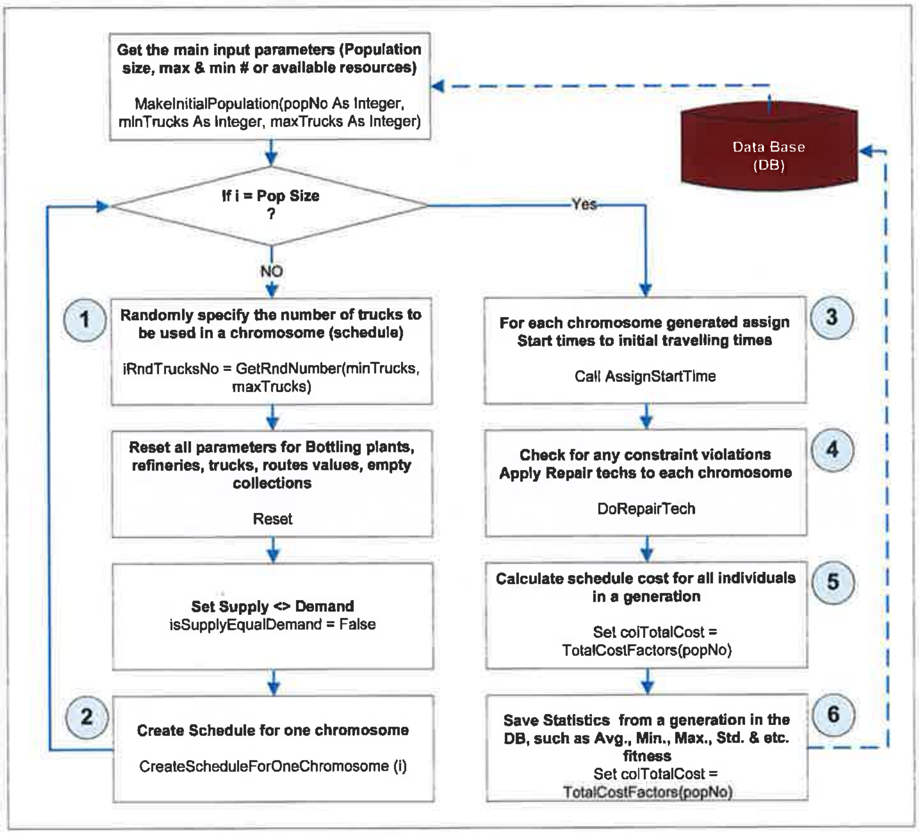


Figure 4-25 Initial Population Generation

4.5.3.1 Resource Assignments

The purpose of this step is to set the total number of trucks to be used in transportation schedule. This is performed by randomly choosing total number of trucks between the specified limits. Figure 4-26, demonstrate the function developed for this purpose.

```
Public Function GetRndNumber(lower As Integer, upper As Integer) As Long
  Dim rndNo As Long
    Randomize
  rndNo = Int((upper - lower + 1) * Rnd() + lower)
  GetRndNumber = rndNo
End Function
```

Figure 4-26, Randomly select total number of trucks

4.5.3.2 Initial Chromosome Generation

Chromosomes are transportation schedules with main components of trucks, routes and start times. The specified number of trucks from the last stage is used to generate a schedule. In this step, the specified routes linking sources to demand centres are randomly assigned to trucks used in the schedule. While the demand and supply are not met this assignments are continued. Finally the generated schedule is stored in the database. This operation is performed by a function called ***CreateScheduleForOneChromosome (i).*** This is a nested function as shown in Figure 4-27, calling other functions to perform this task efficiently and error free. In particular this function calls on other function called ***FillScheduleOneStep(intChrId As Integer)***.

This is a major function developed to create schedules. In this function individual transportation schedule is generated for all the trucks considered for the chromosome. Routes are assigned randomly to each truck. Route assignments are performed, by considering the availability of LPG supply at refineries and LPG demands at bottling plants. Therefore, in each route assignment, the supply capacities and demand levels are updated. In case, where there is not enough LPG available at sources, then the routes leading to that source are removes for further assignment. Accordingly, when bottling plants are full filled with their demands, then the routes ending to this plant are removed for further assignment. This function works based on the assumption that all the demands must be met and also all the supplies must be delivered. Figure 4-28, further demonstrates different steps taken to create chromosomes.

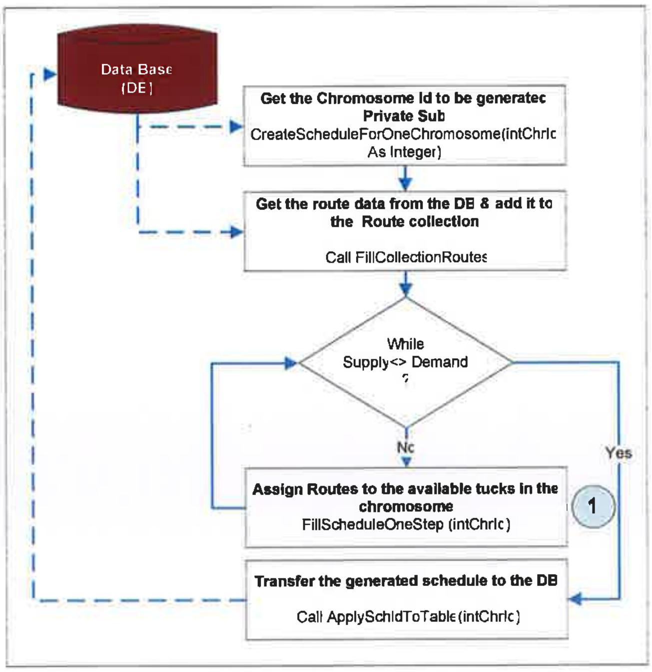


Figure 4-27 creating a chromosome (i.e.schedule)

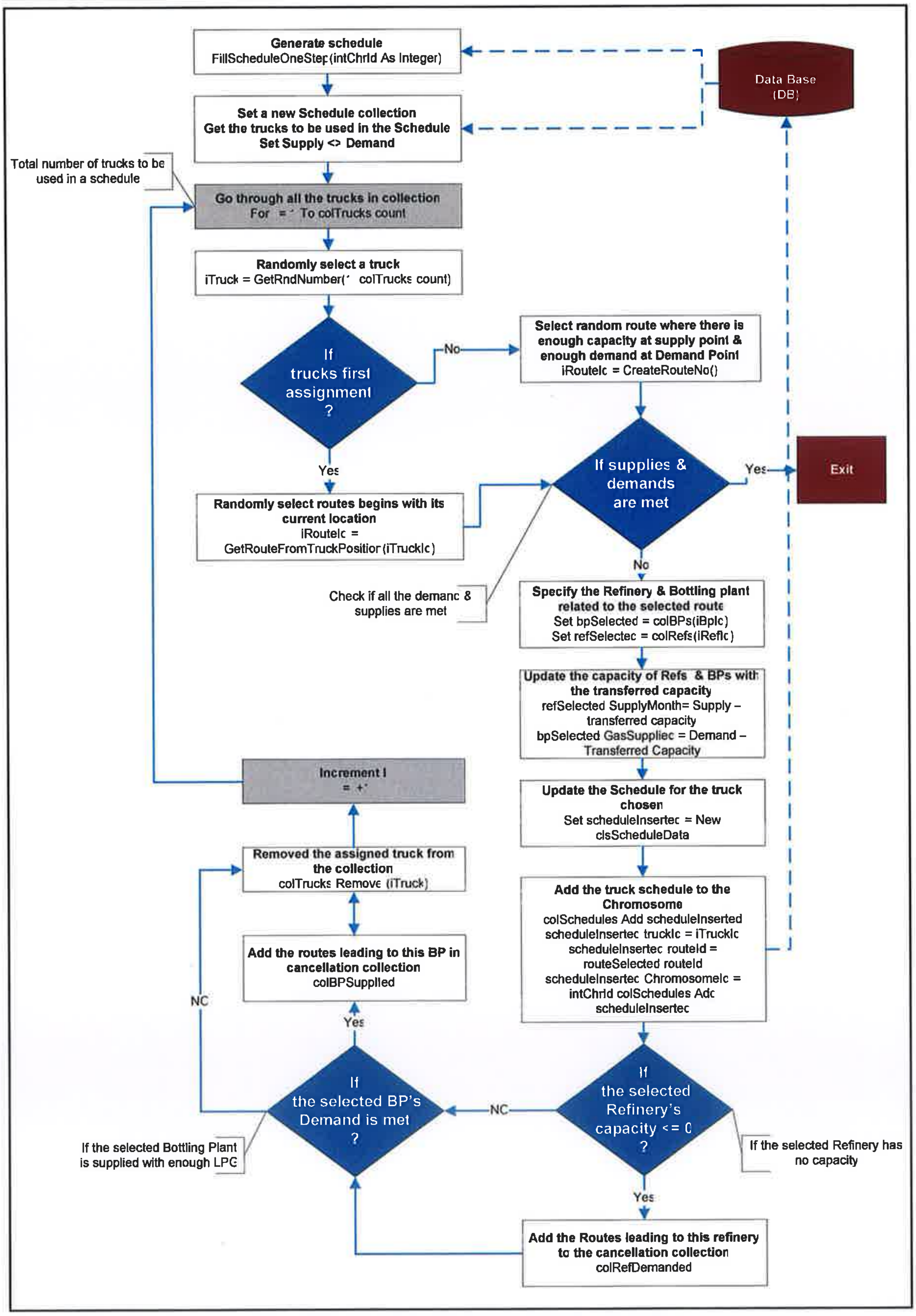


Figure 4-28 FillScheduleOneStep Function

4.5.3.3 Start time Assignments

In this step, a complete schedule is considered. Using a schedule the total number of trucks used is obtained. In addition the routes assigned to each truck is considered. In this step all the routes assigned to trucks are updated with start times. The start time is used to specify when a truck is going to depart a refinery to a bottling plant and also when to depart from a bottling to a refinery. In assigning start times, travelling times, and also possible waiting time at both refineries and bottling plants are considered.

As illustrated in Figure 4-29, the start time for refineries is randomly chosen from a time range of (1 to 24 hours), this reflects the waiting time that trucks are mostly faced at refineries. Also, the start time at bottling plant is randomly set between (1 to 5 hours), which reflects the waiting time that is normally encountered at bottling plants. Therefore, considering the waiting time at supply and demand nodes and the actual travelling time between supply and demand nodes, the departure time for a truck to depart the demand node is the addition of these waiting times and travel times. Figure 4-31 shows the steps taken in function called ***AssignStartTime()*** to update start times for all the considered routes in a chromosome.

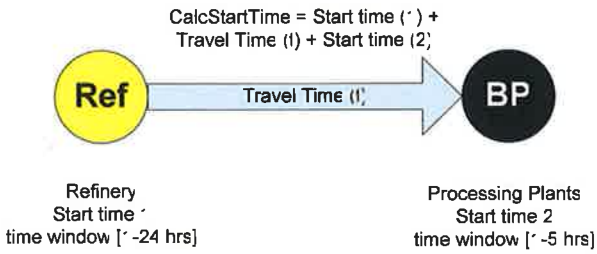


Figure 4-29 Start times calculations

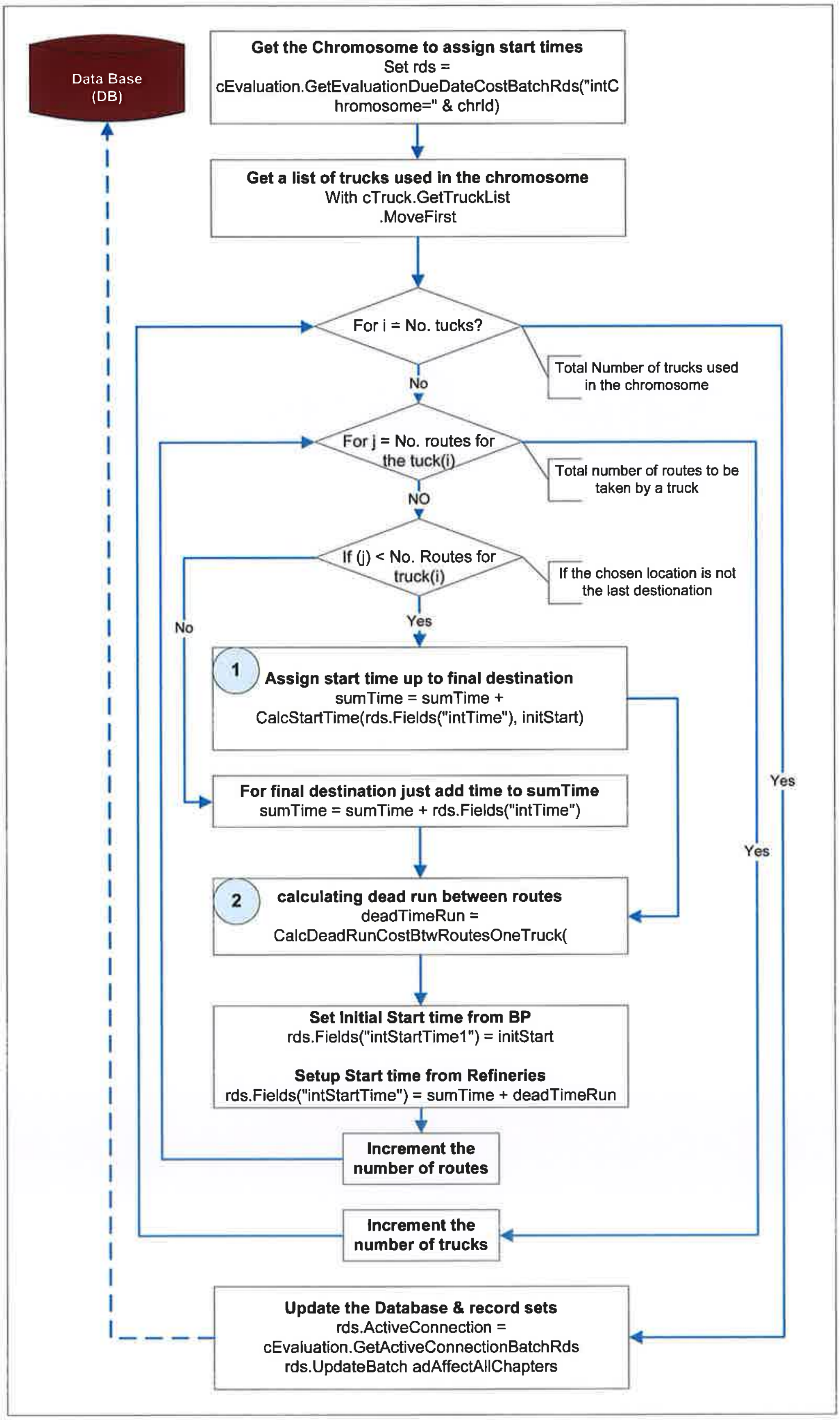


Figure 4-30 Updating chromosomes with start times

4.5.3.4 Repair Technique

This is one of the major steps in developing valid and feasible transportation schedules. As indicated earlier, the logistics systems are typically restricted by a number of real world constraints. Violation of such constraints results in total invalidity of the generated solution. In this application the initial population is set to be composed of all valid solutions. For this reason a set of repair techniques was developed to detect any violations of the specified constraints and then to repair such violations to make valid and feasible solutions. Also, one has to remember that genetic reproduction techniques such as crossover and mutation could extend the solution to infeasible regions and therefore results in infeasibility of the solution. For this reason the repair techniques are applied both after initial population and genetic reproduction processes. The followings briefly describes the developed repair techniques for this application:

1) Supply and Demand Repair Techniques:

This repair technique is used to make sure that the LPG demand and Supply matches the specified limits. In general, there are two conditions for this technique. First and the most common one is to consider the demand and supply must match. This means all the supply limits must be delivered in a way to match all the demands at processing plants. The second option is to consider situations where supply and demands can be variable. This means assignments, where demands are not met or supplies are not totally delivered are acceptable. The aim of this repair tech is to look at each generated schedules and based on the specified demand and supply conditions re-address the violation of such limits.

2) Delivery Operations Repair Technique:

This repair technique was developed based on the lower and upper limits sets for the number of required LPG delivery operations per day. This limit is set for each bottling plant existing in the system. This technique looks at the generated schedules and checks if the specified limits are followed for each available bottling plant. This method establishes the daily deliveries to each bottling plant. If this does not mach the required limits then this techniques tries to address the imbalance between those days with high number of deliveries and lack of deliveries.

3) Distance Repair Technique:

This technique is mainly used to address any constraint violation set for the total distance travelled by the generated schedules. In case of exceeding the required limit on the distance, this repair technique aims to reassign routes so that the total distance is equal or less than the set limit. This could be used for distance minimisation.

4) Human factor Repair Technique:

There are a number of techniques that could be grouped under the human factor repair techniques category. These techniques are developed to address the following constraints:

- **Upper Limit on the Human factor Factor:** this limit is set to make sure that routes taken in a schedule do not exceed an specified upper limit. The task of the developed repair technique is to first highlight those routes that have violated the specified limits. Then the exceeding routes are replaced with the other routes that lead to the same destination.
- **Route Combinations:** there are certain route combinations that are not acceptable by either drivers or unions. The user specifies these route combinations. The aim of the repair technique is to highlight all the violated route combination in a generated schedule and then to replace them with the valid routes.
- **Route Matching:** also, there are certain route combinations that must be followed in a given schedule. The unions specify these route combinations. The repair method must detect if the specified matching routs are met. Otherwise, the violated combinations must be replaced with a valid set.

5) Work balance Repair Techniques:

There are a number of instances of this technique. The first method is based on balancing the number of operations carried out by each truck in a schedule. This method specifies an upper bond on the number of delivery operations to be performed by trucks in a schedule. This limit is obtained based on the total number delivery operations required and the total number of trucks that exist in the schedule. Using this limit, the repair method obtains the total number of delivery operations for each truck in the schedule and then it tries to balance the under utilised trucks with the over utilised ones. This approach makes sure that trucks conduct relatively close number of delivery operations. Although this method was a useful approach to balance the workload but it could overload operators in terms of working hours.

For this reason, the second instance of this repair technique was developed to look at the trucks availability. The availability is usually based on weekly working hours that a truck can operate. Considering the breakdowns and repair times for trucks, one can set this limit. Using this limit, the repair technique gets the total working hours for each tuck in a schedule. Then it balances the under utilised trucks by the over utilised ones. It must be mentioned that none of the trucks are allowed to exceed the specified availability. Therefore, if all trucks are balanced and there are still delivery operations then extra trucks are added to perform such deliveries without violating the availability limits.

6) Completion Time Repair Techniques:

This repair technique is used to make sure that transportation schedules are completed by a specified number of days. The user sets this limit and the repair method highlights those truck schedules violating such limit. The specified truck schedules are reduced to the specified time limit and the extra operations of such trucks are randomly assigned to those trucks that have high gaps to the specified time limit.

4.5.3.5 Chromosome Cost Calculations

This aim of this step is to calculate the overall schedule cost based on the cost parameters and constraint violations described earlier. The algorithm presented in Figure 4-31 uses chromosome details such as the number of truck, routes taken by each truck and the start times to calculate cost measures such as travelled distance, completion time, work balance and etc. The cost is measured for each chromosome in a generation. The cost calculations are detailed in section 4.7.6 and 4.8.

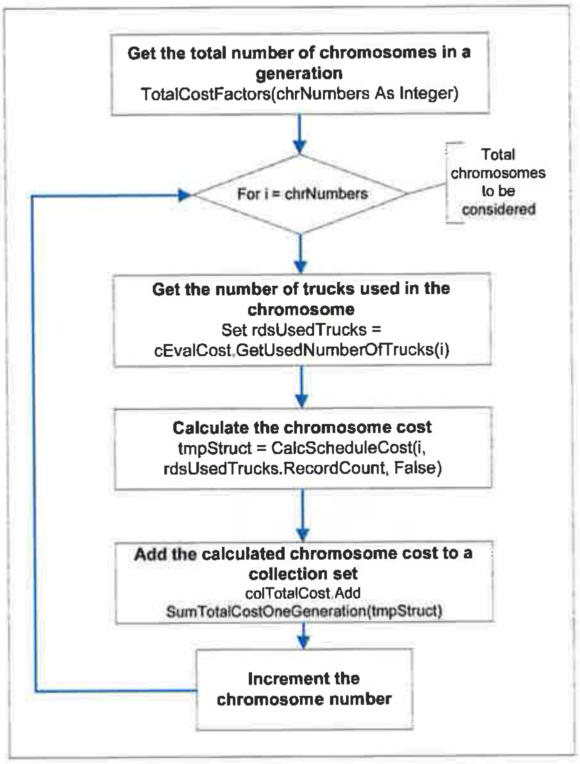


Figure 4-31 chromosome cost calculations

4.5.3.6 Storing Generation into Database

When, all chromosomes in a population are generated and their respective costs (i.e. fitness) are obtained, then details of the generated population is recoded. In this step, four operations take place. First the minimum cost chromosome is identified and then its details such as cost parameters and the number of used trucks are stored. Also, The highest cost schedule is recorded accordingly. Furthermore, a general report on the population is provided. Using this report one can monitor the GA operation performance in terms of convergence rate, selection intensity, population diversity and etc. The generation statistics is recorded in ***Optimum Cost*** database. Finally, the minimum cost individual is compared with other minimum cost individuals from last generation and if the current one is better then details of the schedule are stored in a ***BestChromosome*** Database table. Figure 4-32 shows these steps and Figure 4-33 indicates the related database tables.

```
Private Sub SaveCostToDB()

    Dim iBestChrId As Integer
    Dim iWorstChrId As Integer

'1- Save best cost and return chromosome No                                  1
    iBestChrId = SaveSpecifiedChrCost(iNumberOfGeneration, colTotalCost, True)

'2- Save the most expensive cost                                             2
    iWorstChrId = SaveSpecifiedChrCost(iNumberOfGeneration, colTotalCost, False)

'3- Save the statistics for this generation                                  3
    Call SaveOptimumChromosomeStatistics(iNumberOfGeneration, colTotalCost)
                                                                             4
'4- Store the minmum cost in a DB
    If (iBestChrId > 0) Then
        Call CopyBestChromosome(iBestChrId)
    End If

End Sub
```

Figure 4-32 Save Chromosome cost to DB.

BestChromosome : Table

intSchedule	intTruckId	intRouteId	intChromoso	intStartTime	intStartTime1
454607	13	11	3	1278	0
454608	13	28	3	2169	0
454609	13	39	3	4311	0
454610	13	29	3	4910	0
454611	13	16	3	5841	0
454612	13	9	3	7671	0
454613	13	11	3	10301	0
454614	13	30	3	12881	0
454615	13	10	3	16027	0
454616	13	11	3	17789	0
454617	13	11	3	19427	0
454618	13	11	3	20465	0
454619	13	35	3	21841	0

Record: 1 of 473

MaximumCost : Table

intMaxCostId	intChromosome	intReplicate	dblCost	intTruckNo
45	1	0	628197	24
46	4	0	628440	30
47	4	1	639568	35
(AutoNumber)	0	0	0	0

Record: 3 of 3

MinimumCost : Table

intMinCostId	intChromosome	intReplicate	dblCost	intTruckNo
66	1	0	628197	24
67	2	0	623135	39
68	3	0	608068	34
69	3	1	595409	36
(AutoNumber)	0	0	0	0

Record: 4 of 4

OptimumCost : Table

intOptCostId	intChromosome	intReplicate	dblMaxFitn	dblMinFitn	dblAvgFitness	dblSumFitness	dblVarianceFitn	dblStdDevFitne	dblCost
103	3	0	6.08068	6.2844	6.2196	24.8784	68814714.5	8295.4634892	608068
104	3	1	5.95409	6.39568	6.18017	24.72068	256181457.5	16005.669542	595409
105	1	2	6.03156	6.31978	6.192905	24.77162	140126197.25	11837.491172	603156
(AutoNumber)	0	0	0	0	0	0	0	0	0

Record: 1 of 3

Figure 4-33 Data storage from a population

4.5.4 Populating Generations

The initial population represents the selection pool for chromosomes to be used for populating succeeding generations. The aim here is to populate new generation using the old generation individuals. This is done using Genetic selection and reproduction methods. Figure 4-34 shows this process schematically. In populating the new generation, if elitism is required then a number of chromosomes specified by the user are transferred from the old generation to the new generation. These chromosomes are typically the fittest individuals.

Figure 4-35, represents the overall steps taken to establish a generation of chromosomes. In building a generation, when Elitism is desired, the number of times to replicate for populating a generation is reduced. This is more evident in procedure known as ***DoReplicateMakeGenerations (popNo As Integer)*** as indicated in Figure 4-36. This function starts with transferring the elite individuals to the new generation. Then the remaining individuals are stochastically selected using selection methods. Based on the probability of crossover and mutation the reproduction process takes place and the generated offspring are inserted into the new generation. As indicated in the algorithm these steps are continued until the new generation is completed. Upon the completion the new generation is moved back to the old generation table known as ***Evaluationtable***. Using this table the general generation's performance measures, the best chromosomes, minimum and maximum chromosomes are recorded as explained and illustrated earlier in Figure 4-33.

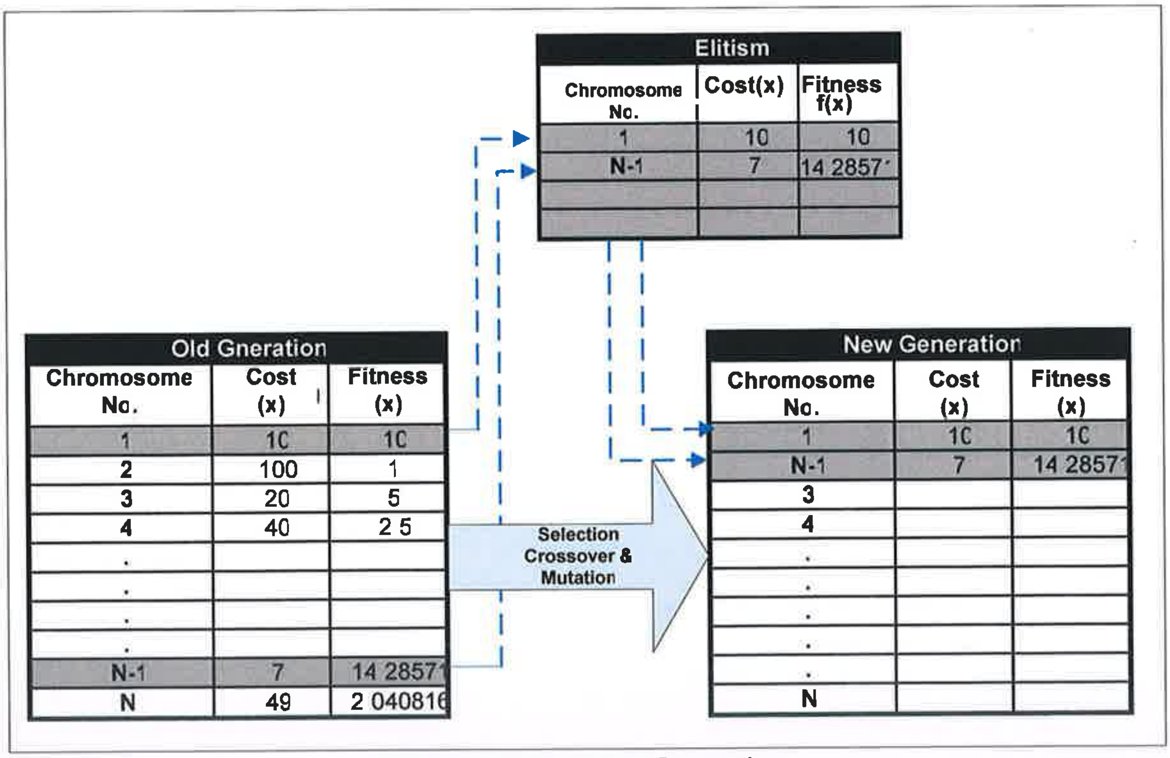


Figure 4-34 Populating Generations

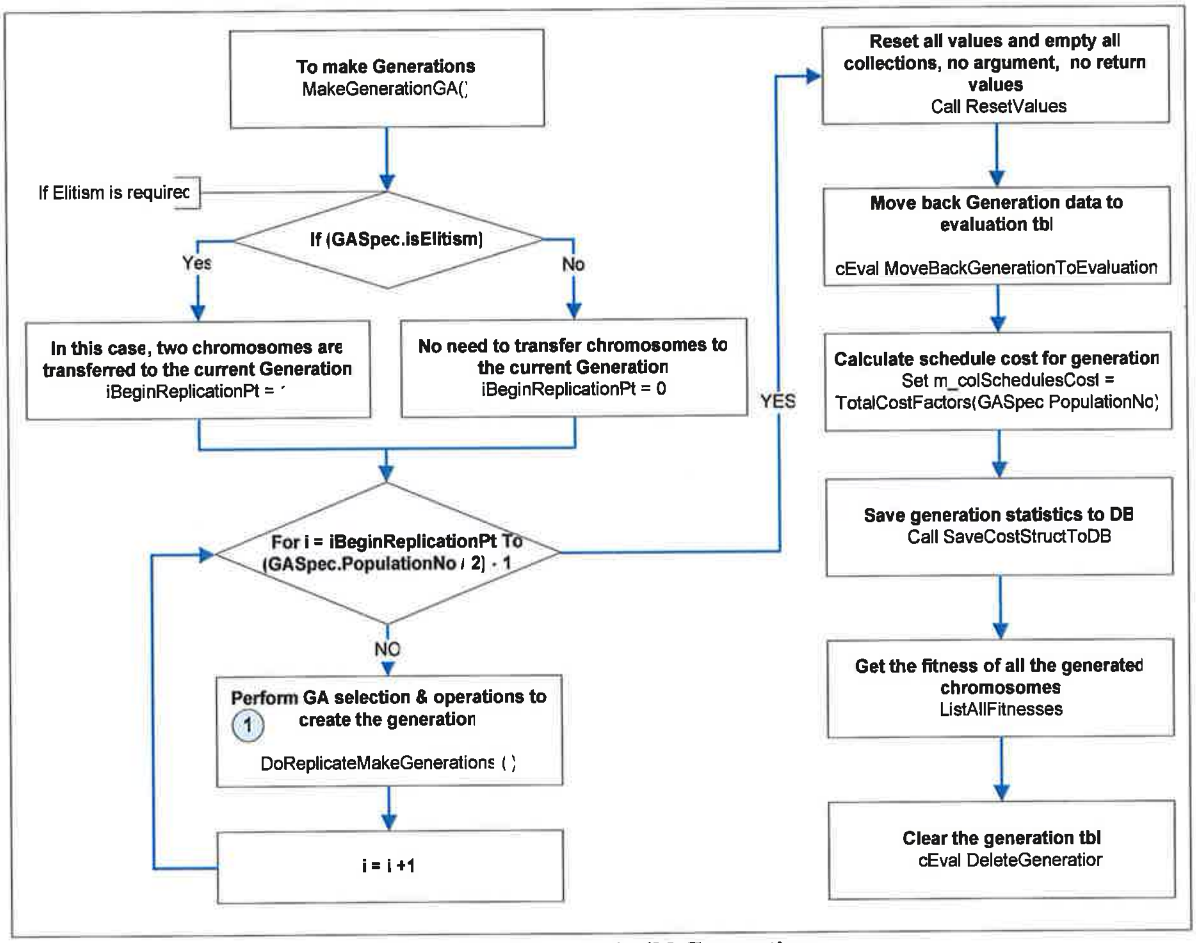


Figure 4-35 Steps to build Generations

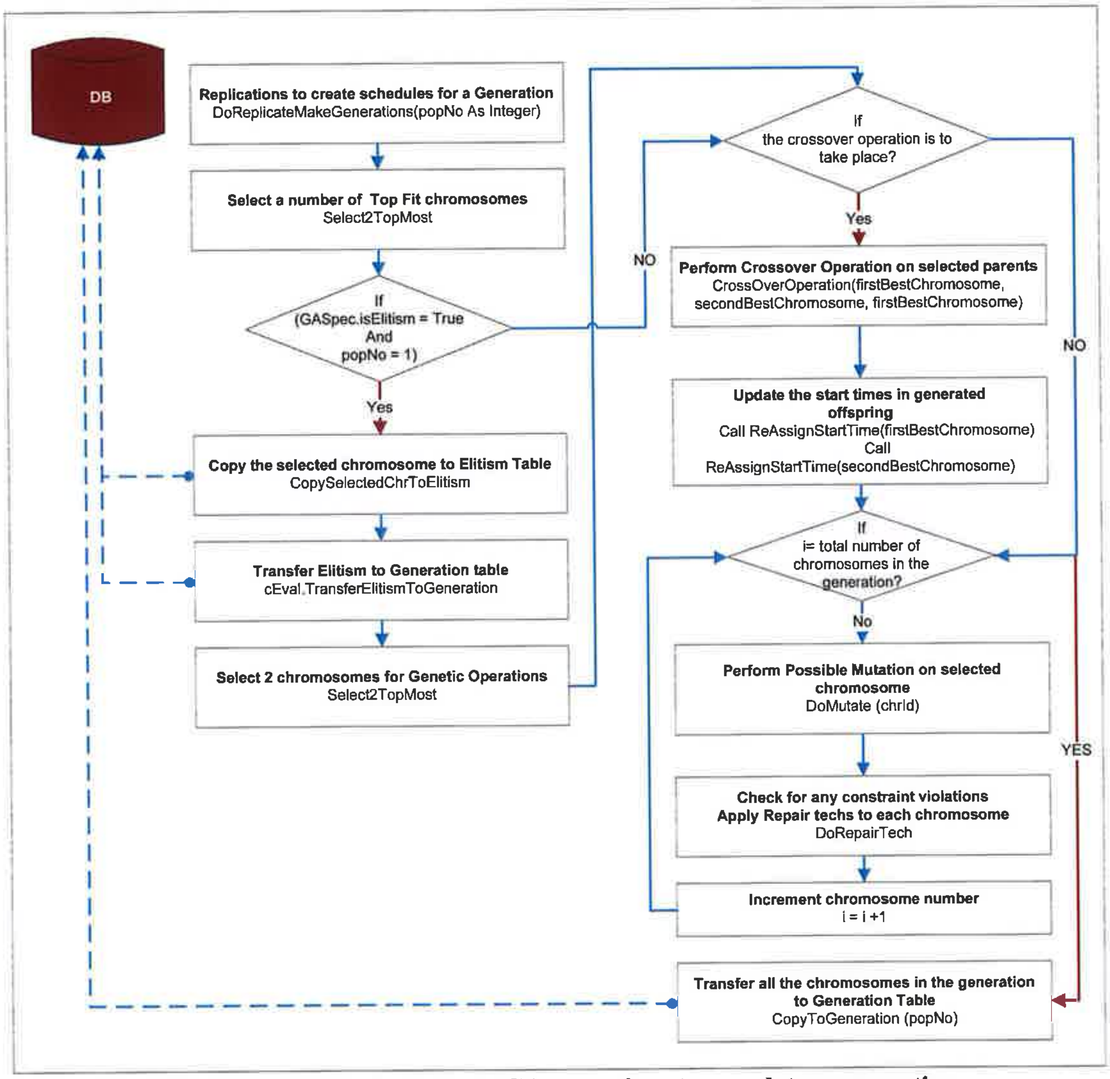


Figure 4-36 Steps to replicate GA operations to populate a generation

4.5.5 Scaling Methods

Prior to selection of parents for reproductions, there is a need of fitness scaling to prevent any possible premature convergence. The convergence to a local minimum could happen both in the early and later runs of the GA operations. At the start of GA runs, it is common to have a few extraordinary chromosomes in a population, which would take over a significant proportion of the population in a single generation. Also, late in the run, the population average fitness may be close to the population best fitness, which may result the average members and the best members to get nearly the same number of copies in future generations. There are a number of scaling methods available in literatures such as windowing, exponential and etc. There are three scaling approached conducted in this application. Here one of the approaches is introduced and descriptions on the linear transformation and the ranked based methods are provided in Appendix C.

4.5.5.1 Linear scaling to values between 0 and 1.

This scaling scheme was implemented in the computer program GENETIC [51]. In this method the raw fitness is defined as f and the scaled fitness as f'. In this approach the highest fitness value f_h and the lowest value f_l of the chromosomes are obtained. The fitness values are converted to positive values by adding the quantity C = 0.1 f_h -1.1 f_l to each of the fitness values. Thus the new highest value will be 1.1[f_h - f_l] and the lowest value 0.1[f_h - f_l]. Each of the new values is then divided by $D = \max(1, f_h + C)$

$$f_i' = {f_i + C}/{D}, \text{Where} \qquad \text{.............} (34)$$

$$C = 0.1 f_h - 1.1 f_l \qquad \text{.............} (35)$$

$$D = \max(1, f_h + C) \qquad \text{.............} (36)$$

After performing the scaling operation, the sum of the scaled fitness values S is calculated using $S = \sum_{i=1}^{z} f_i'$. In order to demonstrate the impact of such scaling mechanism, this method is applied to the fitness values obtained earlier in section 4.5.2. Table 4-2, illustrates how the 0-1 scaling method is applied to the fitness values. In addition, Figure 4-37 illustrates how this method was coded in this application. Finally, looking at Figure 4-38 illustrates the impact of scaling method. It is evident that less fit chromosomes such as chromosome 9, 1 and 3 are now given better weight and therefore better probability of selection for genetic reproduction process.

Table 4-2 Scaling Fitness values using (0-1) Method

Chromosomes	Initial Costs	Fitness Value (1)	(1)+C/D	Sum of Fitness	Fitness Value (1) Cost Ratio%	Scaled Fitness Value (1) Cost Ratio%
1	178781	6758	0.13617	0.13617	1	2
2	132979	52560	0.44300	0.57917	7	7
3	158753	26786	0.27034	0.84951	4	5
4	79398	106141	0.80193	1.65143	14	13
5	84155	101384	0.77006	2.42150	13	13
6	78149	107390	0.81029	3.23179	14	14
7	49830	135709	1.00000	4.23179	18	17
8	73345	112194	0.84248	5.07427	15	14
9	185539	1	0.09091	5.16518	0	2
10	80323	105216	0.79573	5.96091	14	13

Lowest Cost	FL	1
Highest Cost	Fh	135709
	C	13569.80
	D	149278.80

```
Private Function ScalingNormalMethod() As Double

   Dim i As Integer
   Dim sumTotalFitnesses As Double
   Dim QuantityC As Double
   Dim QuantityD  As Double
   Dim beforeScaled As Double
   Dim afterScaled As Double

   ' calc QuantityC, QuantityD
   Call GetQuantityVal(QuantityC, QuantityD)

   Dim valElement As Variant

   ' calc sum of all fitness values
   For Each valElement In dicFitness.Keys
      beforeScaled = CDbl(dicFitness.Item(valElement))

      afterScaled = NormalScalingFitness(beforeScaled, QuantityC, QuantityD)
      dicScaledFitness.Add valElement, afterScaled

      sumTotalFitnesses = sumTotalFitnesses + afterScaled
   Next

   ScalingNormalMethod = sumTotalFitnesses

End Function
```

Figure 4-37, 0-1 Scaling method

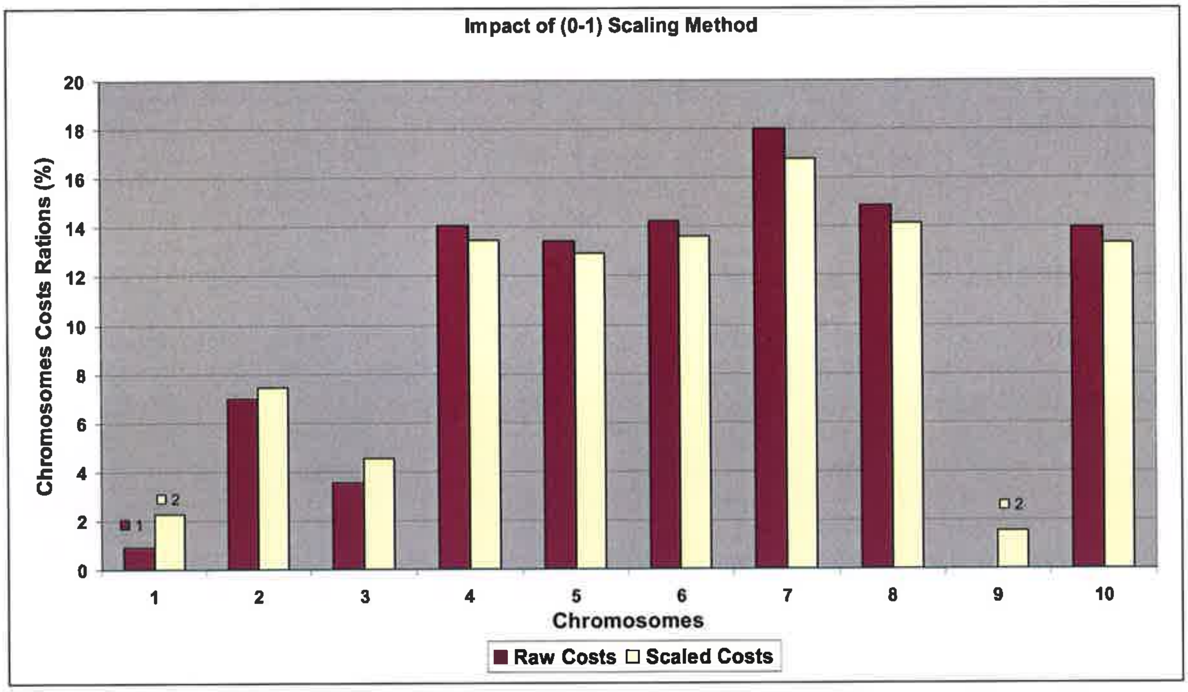


Figure 4-38 Impact of Scaling on chromosomes fitness

4.5.6 Selection Methods

One of the main GA operators that would influence the performance of the evolutionary algorithm is the selection method. Chromosomes are selected from the population to be parents for the reproduction process, the fitter the chromosome, the more times it is likely to be selected to reproduce. The approaches implemented in this work are roulette wheel and tournament selection methods. The latter method is presented in Appendix C.

4.5.6.1 Roulette Wheel:

Figure 4-39, demonstrates how this method operates. First, the total population fitness is calculated and then a random number ranging from 0 to the total population fitness is generated. Starting from the first chromosome, the fitness values are summed and if the fitness values exceed the generated random number then that chromosome is selected for reproduction process.

This selection method is in fact a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness value. This method chooses chromosomes probabilistically, instead of deterministically. In this way, if a chromosome has the highest fitness there would be no guarantee that it will be selected. In fact, on average a chromosome will be chosen with the probability proportional to its fitness. Figure 4-40, illustrates how this method was coded in this application.

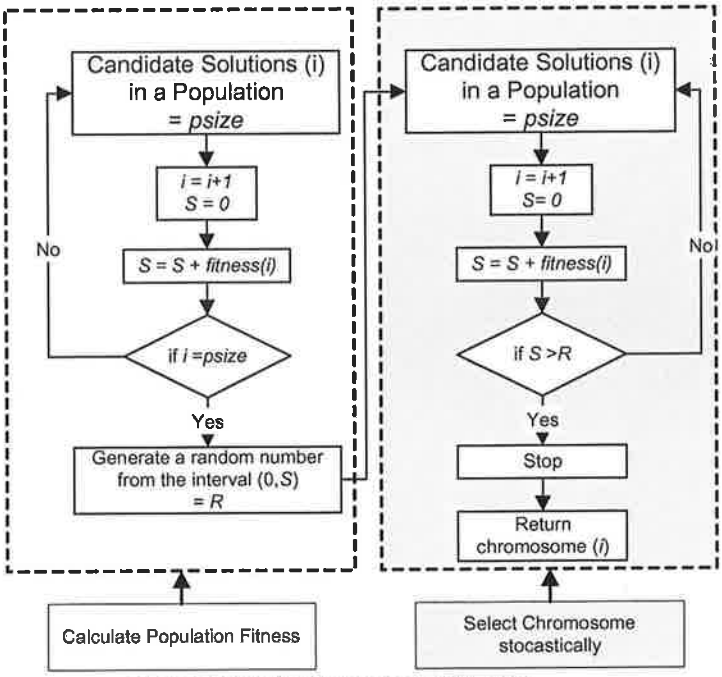


Figure 4-39 Roulette wheel selection

```
Private Function SelectionNormalMethod(sumTotalFitness As Double) As Integer

   Dim dblInterval As Double
   Dim sumFitness As Double
   Dim tmpNumber As Double
   Dim i As Integer
   Dim count As Integer

   ' Generate interval number
   dblInterval = GetRndNumberWithFigures(0, 1)
   tmpNumber = sumTotalFitness * dblInterval
   sumFitness = 0
   count = -1
   ' Get the highest & lowset
   For i = 0 To (dicScaledFitness.count) - 1
      ' do add fitness
      sumFitness = sumFitness + dicScaledFitness.Item(dicScaledFitness.Keys(i))
      ' if less than interval's fitness, continue, otherwise, exit
      If (sumFitness > tmpNumber) Then
         count = i
         Exit For
      End If
   Next

   ' return the selected chromosome
   SelectionNormalMethod = dicScaledFitness.Keys(count)

End Function
```

Figure 4-40, Roulette wheel function

4.5.7 Crossover Operators

Crossover serves as a mechanism by which parent chromosomes can exchange information and possibly creation of more fit offspring. This allows exploration of new regions of the search space. There are a number of operators developed in this application. Here, only one operator is described and Appendix C provides further details on the other operators. To demonstrate the impact of these operators the following two transportation schedules are considered.

- **Schedule 1:**

Cost	S1	CL	SL	EL	SL	EL	SL	EL	SL	EL	SL	EL
79617			27	27	15	15	19	19	28	28	15	15
7695	T1	Babol	Esfahan	Karaj	Abas	Qom	Arak	Karaj	Esfahan	Kashan	Abas	Qom
			12	12	14	14	11	11	31	31		
13675	T2	Esfahan	Abas	Kashan	Abas	Mashhad	Abas	Karaj	Esfahan	Qom		
			35	35	12	12	18	18	9	9	11	11
15765	T3	Karaj	Tehran	Karaj	Abas	Kashan	Arak	EsfahanC	Abas	Babol	Abas	Karaj
			15	15	7	7	19	19	14	14		
15150	T4	Kashan	Arak	Qom	Arak	EsfahanC	Arak	Mashhad				
			14	14	15	15	15	15	40	40		
14201	T5	Kerman	Abas	Mashhad	Abas	Qom	Abas	Qom	Tehran	yazd		
			12	12	22	22	24	24	11	11		
13131	T6	Mashhad	Abas	Kashan	Arak	Mashhad	Arak	yazd	Abas	Karaj		

Figure 4-41 a chromosome representing a schedule

Figure 4-41, shows a chromosome representing a transportation schedule for six trucks. Each individual truck has a unique schedule, which consists of different location to travel. Also the schedule cost for each individual truck as well as the overall transportation schedule is recorded as shown in this figure.

- **Schedule 2:**

Cost	S2	CL	SL	EL	SL	EL	SL	EL	SL	EL	SL	EL
163908			38	38	13	13						
19763	A	Esfahan	Tehran	Mashhad	Abas	Kerman						
			3	3	39	39	17	17				
11496	B	Tehran	Abadan	Karaj	Tehran	Qom	Arak	Babol				
			4	4	21	21	27	27				
6738	C	Babol	Abadan	Kashan	Arak	Kerman	Esfahan	Karaj				
			15	15	7	7	19	19	14	14		
13490	D	Esfahan	Abas	Qom	Abadan	Qom	Arak	Karaj	Abas	Mashhad		
			29	29								
10054	E	Karaj	Esfahan	Kerman								
			2	2	14	14	14	14				
16275	F	Kashan	Abadan	EsfahanC	Abas	Mashhad	Abas	Mashhad				
			3	3	8	8	35	35	10	10		
28821	G	Kerman	Abadan	Karaj	Abadan	yazd	Tehran	Karaj	Abas	EsfahanC		
			26	26	11	11	37	37				
8400	H	Mashhad	Esfahan	EsfahanC	Abas	Karaj	Tehran	Kerman				
			25	25	7	7	38	38	22	22		
12380	I	Qom	Esfahan	Babol	Abadan	Qom	Tehran	Mashhad	Arak	Mashhad		
			23	23	12	12	33	33				
17960	J	Yazd	Arak	Qom	Abas	Kashan	Tehran	Babol				
			33	33	26	26	17	17	10	10		
6406	K	Abadan	Tehran	Babol	Esfahan	EsfahanC	Arak	Babol	Abas	EsfahanC		
			13	13	2	2	21	21	19	19	35	35
10125	L	Abas	Abas	Kerman	Abadan	EsfahanC	Arak	Kerman	Arak	Karaj	Tehran	Karaj

Figure 4-42 A chromosome representing a transportation schedule

Figure 4-42, similarly illustrates another chromosome containing plans for twelve trucks. Each of these resources has their transportation plan and their respective costs as illustrated. The following sections use these two chromosomes to perform the crossover operations. Figure 4-43, shows the overall scheme in applying different crossover operation used within this application.

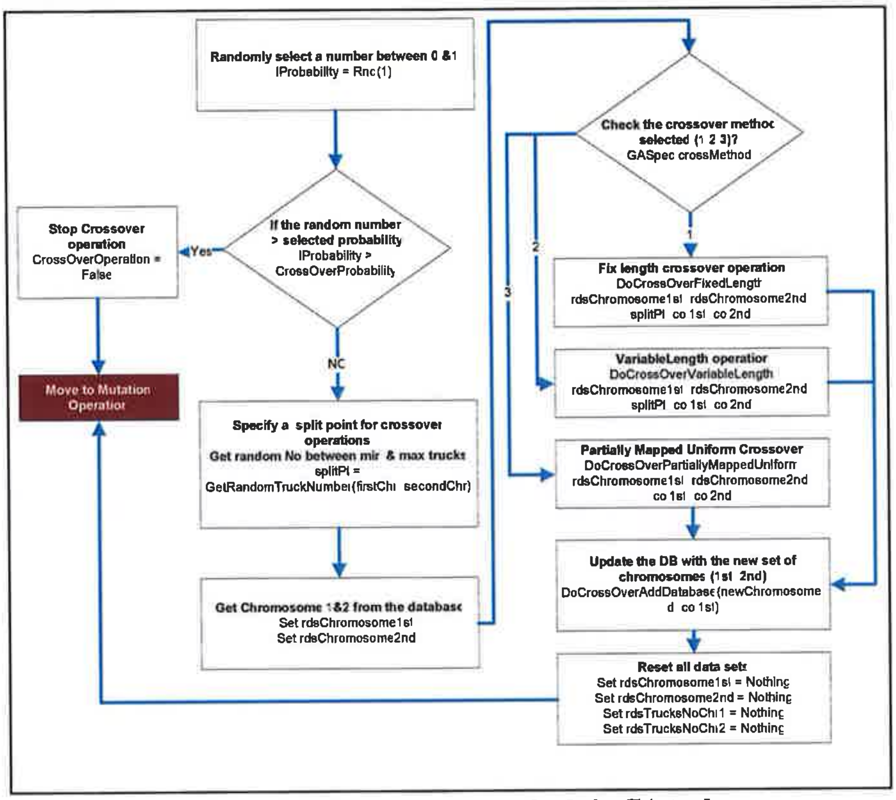


Figure 4-43 the crossover methods in GA tool

4.5.7.1 Single Point, Fixed Length Crossover:

This operator is used to generate an offspring with the same length as the parent chromosomes. Here the length refers to the number of trucks considered in a schedule. The idea of this operator is to keep the number of tuck fixed in successive generations. Thereby a crossover point is randomly selected between the minimum and maximum length of the shortest chromosome (e.g. 5). The first portion of the parent genes remains the same and the second portions are exchanged as shown as follows.

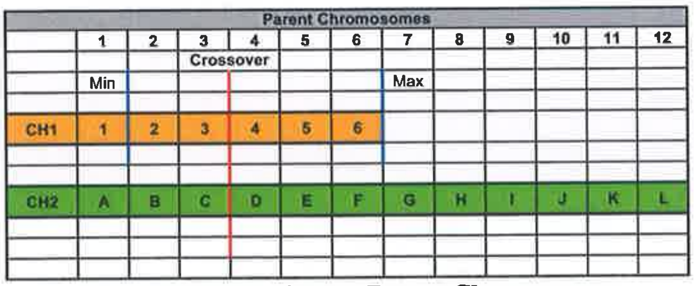


Figure 4-44 Top view on Parent Chromosomes

Figure 4-44, illustrates a top view of the described chromosomes. As indicated before, the aim of this operation is to perform crossover while keeping the total number of trucks consistent with the parents. For this reason the chromosome containing the least number of trucks is chosen. The crossover point is randomly chosen between the minimum and the maximum number of trucks used in this schedule.

Offspring Chromosomes												
	1	2	3	4	5	6	7	8	9	10	11	12
Ofs1	1	2	3	D	E	F						
Ofs2	A	B	C	4	5	6	G	H	I	J	K	L

Figure 4-45 Generated offspring

Figure 4-45 illustrates how the first segment of each parent chromosomes are kept and unchanged and also the number of trucks used in the offspring stays the same as in the parents.

Cost	S1	CL	SL	EL	SL	EL	SL	EL	SL	EL	SL	EL
78954			27	27	15	15	19	19	28	28	15	15
7695	T1	Babol	Esfahan	Karaj	Abas	Qom	Arak	Karaj	Esfahan	Kashan	Abas	Qom
			12	12	14	14	11	11	31	31		
13675	T2	Esfahan	Abas	Kashan	Abas	Mashhad	Abas	Karaj	Esfahan	Qom		
			35	35	12	12	18	18	9	9	11	11
15765	T3	Karaj	Tehran	Karaj	Abas	Kashan	Arak	EsfahanC	Abas	Babol	Abas	Karaj
			15	15	7	7	19	19	14	14		
13490	D	Esfahan	Abas	Qom	Abadan	Qom	Arak	Karaj	Abas	Mashhad		
			29	29								
10054	E	Karaj	Esfahan	Kerman								
			2	2	14	14	14	14				
18275	F	Kashan	Abadan	EsfahanC	Abas	Mashhad	Abas	Mashhad				

Figure 4-46 Schedule 1 after crossover

Figure 4-46 and Figure 4-47, indicate offspring generated after crossover resulting in a new set of schedules. As evident from Figure 4-47, a better schedule is obtained representing a lower cost. Figure 4-48, shows how this method implemented in this application.

Cost	S2	CL	SL	EL	SL	EL	SL	EL	SL	EL	SL	EL
164571			38	38	13	13						
19763	A	Esfahan	Tehran	Mashhad	Abas	Kerman						
			3	3	39	39	17	17				
11496	B	Tehran	Abadan	Karaj	Tehran	Qom	Arak	Babol				
			4	4	21	21	27	27				
6738	C	Babol	Abadan	Kashan	Arak	Kerman	Esfahan	Karaj				
			23	23	18	18	22	22				
15150	T4	Kashan	Arak	Qom	Arak	EsfahanC	Arak	Mashhad				
			14	14	15	15	15	15	40	40		
14201	T5	Kerman	Abas	Mashhad	Abas	Qom	Abas	Qom	Tehran	yazd		
			12	12	22	22	24	24	11	11	10	
13131	T6	Mashhad	Abas	Kashan	Arak	Mashhad	Arak	yazd	Abas	Karaj	Abas	
			3	3	8	8	35	35	10	10		
28821	G	Kerman	Abadan	Karaj	Abadan	yazd	Tehran	Karaj	Abas	EsfahanC		
			26	26	11	11	37	37				
8400	H	Mashhad	Esfahan	EsfahanC	Abas	Karaj	Tehran	Kerman				
			25	25	7	7	38	38	22	22		
12380	I	Qom	Esfahan	Babol	Abadan	Qom	Tehran	Mashhad	Arak	Mashhad		
			23	23	12	12	33	33				
17960	J	Yazd	Arak	Qom	Abas	Kashan	Tehran	Babol				
			33	33	26	26	17	17	10	10		
6406	K	Abadan	Tehran	Babol	Esfahan	EsfahanC	Arak	Babol	Abas	EsfahanC		
			13	13	2	2	21	21	19	19	35	35
10125	L	Abas	Abas	Kerman	Abadan	EsfahanC	Arak	Kerman	Arak	Karaj	Tehran	Karaj

Figure 4-47Schedule 2 after crossover

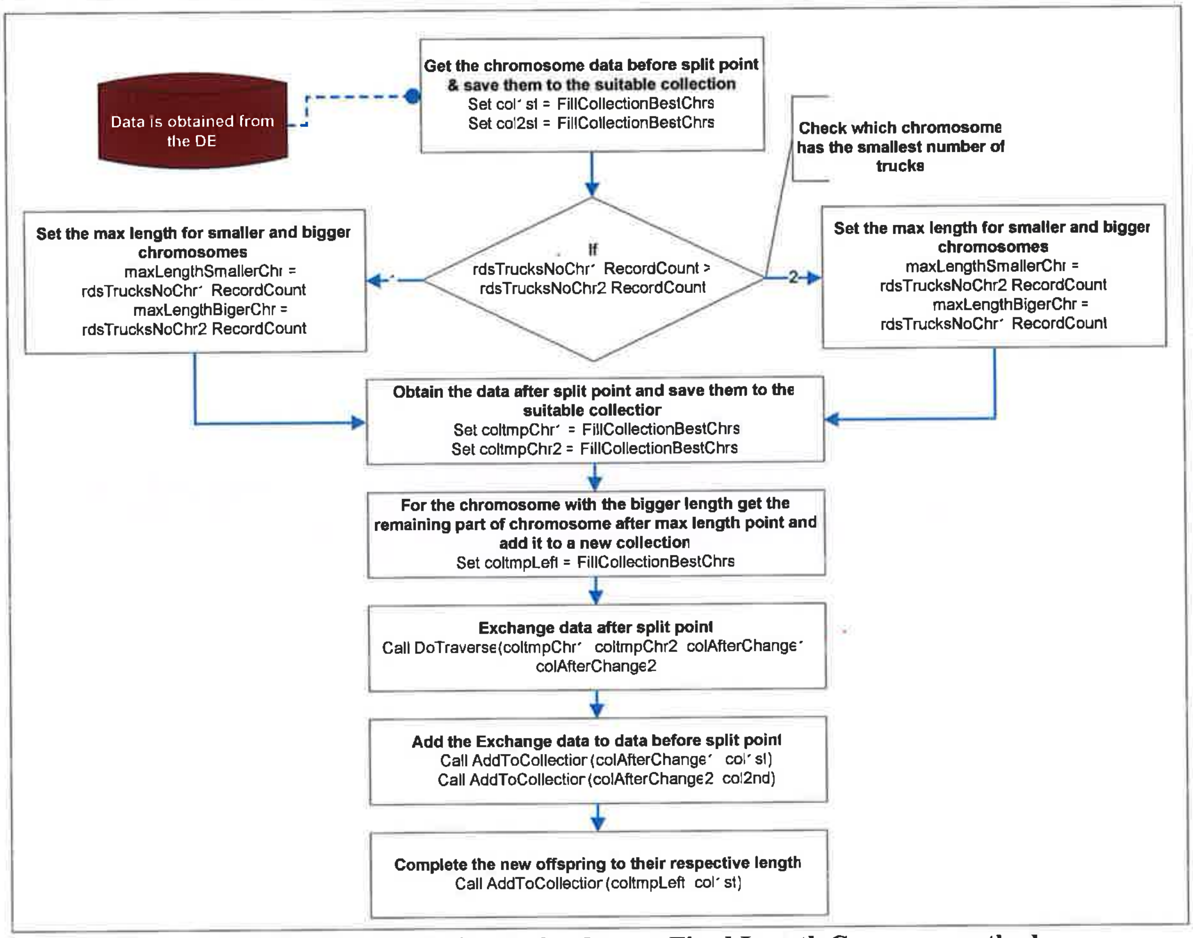


Figure 4-48 series of functions to implement Fixed Length Crossover method

4.5.8 Mutation Operators

Mutation is a random exchange of one or more genes in an offspring resulted from the crossover operation. The primary purpose of this operator is to increase diversity into a population. The mutation may result in finding better optima. Figure 4-49 illustrates how mutation operators were implemented in this application. In this application two types of mutation operators were developed. The Classic method is described here and the Inversion method is illustrated in Appendix C.

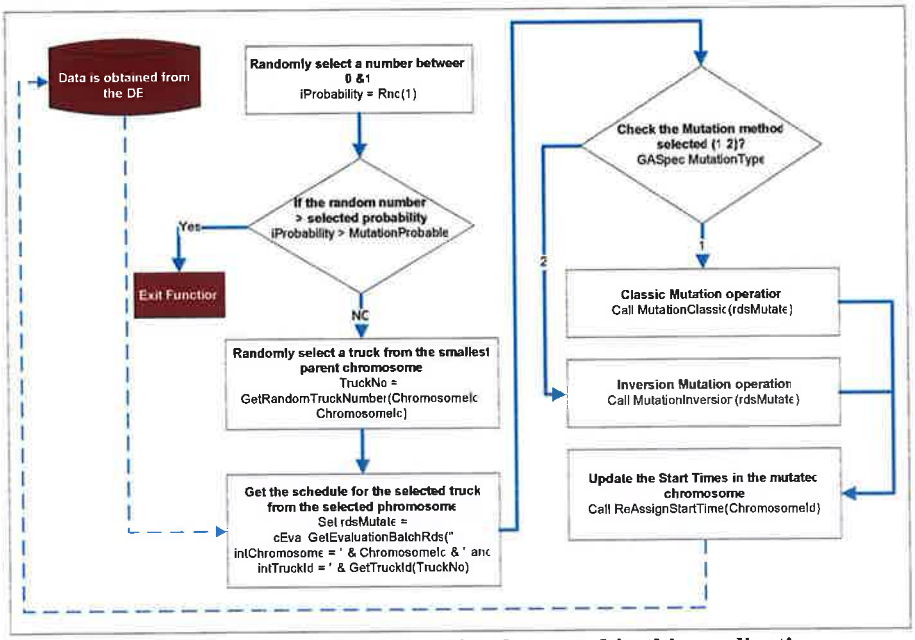


Figure 4-49 Mutation Methods implemented in this application

4.5.8.1 Classic Mutations:

This operator randomly selects a truck schedule from a generated offspring. The detail of the schedule is such as routes and start times are replaced with a new set of random assignment of such variables. Figure 4-52, illustrates how this operator was coded within this application. This operator is demonstrated as follows:

Cost	S2	CL	SL	EL	SL	EL	SL	EL	SL	EL
67348			38	38	13	13				
19763	A	Esfahan	Tehran	Mashhad	Abas	Kerman				
			3	3	39	39	17	17		
11496	B	Tehran	Abadan	Karaj	Tehran	Qom	Arak	Babol		
			4	4	21	21	27	27		
6738	C	Babol	Abadan	Kashan	Arak	Kerman	Esfahan	Karaj		
			23	23	18	18	22	22		
15150	T4	Kashan	Arak	Qom	Arak	EsfahanC	Arak	Mashhad		
			14	14	15	15	15	15	40	40
14201	T5	Kerman	Abas	Mashhad	Abas	Qom	Abas	Qom	Tehran	yazd
			12	12	22	22	24	24	11	11
13131	T6	Mashhad	Abas	Kashan	Arak	Mashhad	Arak	yazd	Abas	Karaj

Figure 4-50 a chromosome before mutation operation

Cost	S2	CL	SL	EL	SL	EL	SL	EL	SL	EL
77548			38	38	13	13				
19763	A	Esfahan	Tehran	Mashhad	Abas	Kerman				
			3	3	39	39	17	17		
11496	B	Tehran	Abadan	Karaj	Tehran	Qom	Arak	Babol		
			4	4	21	21	27	27		
6738	C	Babol	Abadan	Kashan	Arak	Kerman	Esfahan	Karaj		
			23	23	18	18	22	22		
15150	T4	Kashan	Arak	Qom	Arak	EsfahanC	Arak	Mashhad		
			14	14	15	15	15	15	40	40
14201	T5	Kerman	Abas	Mashhad	Abas	Qom	Abas	Qom	Tehran	yazd
			31	31	31	31	9	9	9	9
10200	T6	Qom	Esfahan	Qom	Esfahan	Qom	Abas	Babol	Abas	Babol

Figure 4-51 Mutated Chromosome

```
Private Sub MutationClassic(ByRef rds As ADODB.Recordset)
    Dim i As Integer
    Dim cRoute As New clsRoute
    Dim iRoute As Integer
    ' Get all routes from DB
    cRoute.GetRouteRds
    With rds
        'Randomly replace all the route assigned to the selectec truck
        '1 for first location of truck to the final destination
        For i = 1 To .RecordCount - 1
            If (.EOF) Then Exit For
            'Randomly select routes from the available routes
            iRoute = GetRndNumber(1, cRoute.GetRouteList.RecordCount)
            'Get the Route data from the database
            Call cRoute.GetRouteList.Find("intRouteId=" & iRoute, , adSearchForward,
adBookmarkFirst)
            'Update the db record for the selected truck with the new route
            If (cRoute.GetRouteList.EOF = False) Then
                .Fields("intRouteId") = cRoute.GetRouteList.Fields("intRouteId")
            End If
            .MoveNext
        Next
        'Update the mutated chromosome in the database
        .ActiveConnection = cEval.GetActiveConnectionBatchRds
        .UpdateBatch adAffectAllChapters
        Set .ActiveConnection = Nothing
    End With
    ' destruct obj
    Set cRoute = Nothing
End Sub
```

Figure 4-52, Classic Mutation operator

4.5.9 Replacement Strategies

When creating new population by genetic reproduction processes, losing the fittest chromosomes is probable as the selection of candidate solutions is done stochastically. In this application the concept of Elitism model was used to allow transfer of a few best chromosomes to new population for further evaluation. In this application, the user can specify any number of elite members to be transferred between generations. Starting from initial population, a number of elite individuals are selected from this population. These elite members are transferred to the next generation. The remaining individuals for this generation are produced from the chromosomes in the last generation using the genetic reproduction methods.

Again, the elite members are updated. In this way each elite member is individually compared with the chromosome in the current generation. If there are chromosomes that are cheaper or fitter than the elite members then the member is replaced by the fitter chromosome. As before, the updated elite members are transferred to the next generation and the remaining individuals are created from chromosomes in the last generation.

The above steps are carried out in a function called ***MakeEliteSchedule(ByRef colCost As Collection)***. This function is a complex nested function that calls on many other developed functions to record and transfer specified number of elite members to succeeding generations. Appendix C, illustrates how this function was developed.

4.6 Pareto Optimal Genetic Search Engine

To perform search based on Pareto Optimal Genetic algorithm (POGA), the user need to specify the same initial parameters as the standard GA search method, including, the initial populations size, the maximum number of trucks, number of generations, scaling method, selection method, crossover type, crossover probability, mutation type, mutation probability, fitness evaluation method. In addition to these parameters the user has the option to choose any combination of objectives to either minimise or maximise during the evolution process. This is illustrated in Figure 4-53. The POGA aims to find the set of non-dominated solutions in a decision space comprised of decision variables such as Trucks, Routes and Start times for an objective function comprised of 9 components. Chapter 3 described different steps involved in this algorithm. In this section the aim is to revisit those steps to show how these were implemented. As this search method needs further modifications and it is not completed. The reader is asked to refer to Appendix C for detail information on how main aspects of this approach were developed in this work.

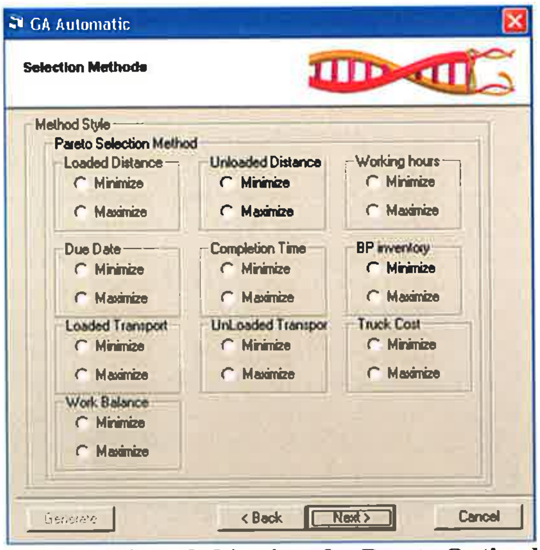


Figure 4-53 Selection of objectives for Pareto Optimal search

4.7 Cost Factors within the Fitness function

The selection of chromosomes in the simple GA is directly based on the fitness of the chromosomes. The fitness of chromosomes is a measure of the quality of solution represented by the chromosome. The probability of a particular individual passing its genes into the next generation was directly proportional to its fitness. The fitness of the chromosomes in this application is based on different cost factors considered in modelling vehicle routing and scheduling. The following sections describe each of these parameters and how they are measured for this problem.

4.7.1 Completion time cost

This cost factor reflects the total time taken to complete all the required transportation to the bottling plants. As the transportation schedules are generated based on monthly demands then a typical time horizon to complete the LPG transportation is about 31 days or less. A generated schedule is evaluated to check when all the deliveries are completed. This time is taken as the completion time. The completion time cost is mainly related to two main cost parameters such as Labour and Transit Inventory costs. Each of these parameters is calculated as follows:

- **Labour cost:**

In calculating the labour cost, there are two main rates considered, namely the standard and overtime rate as indicated in Table 4-3. The earlier rate justifies the transportation cost within the specified time limit and the later is the cost rate applied when the delivery times exceeds this limit. In calculating the labour cost, the overtime-cost rate is projected to indicate the extra labour cost that the company must pay due to the late deliveries. To calculate the labour cost, the total transportation time for each existing truck in the schedule is obtained. If this time lies within the time limit then the standard rate is applied else the overtime rate is considered.

Table 4-3 labour cost calculations

Time Horizon	25
Standard Working Hour/day	18
Standard Labour Rate (/hrs)	31.05
Overtime Labour Rate (/hrs)	50

Trucks	Total Travel Time (min) (1)	Standard working time(min) (2)	Standard Labour Cost if(1>=2) then (2)*standard Rate, Else (1<2) then (1)* Standard Rate	Overtime Labour Cost if(1>2) then (1-2)^Overtime Rate, Else overtime=0	Total Labour cost per truck
1	720	450	13972.5	279.92	14252.42
2	480	450	13972.5	170.06	14142.56
3	840	450	13972.5	298.31	14270.81
4	360	450	11178	0.00	11178.00
5	960	450	13972.5	311.72	14284.22
				Total Schedule labour Cost	68128.01

- **LPG Transit inventory cost:**

Transit inventory refers to the amount of LPG that is in transportation and it is not delivered to processing plants yet. In this calculation the total transit inventory, which is the summation of inventories in each route is recorded on daily bases. Here, the transit inventory is calculated based on the revenue cost. The transit inventory represents the amount of LPG that are not delivered to the final customer and therefore they indicate lost in sales and more importantly lost of customer good will. Table 4-5, illustrates how transit inventory is recorded for a bottling plant in the existing system. Also as indicated in Table 4-4, the transit inventory is calculated for each processing plant. The transit inventory multiplied by revenue cost is the total cost. In this approach, the company used an exponential function to calculate this cost. The intensity level is used to consider factor such as seasonal demand for LPG.

$$TransitCost = \mathrm{Re}\,venue\cos t \times TransitInventory^{IntensityLevel} \quad \ldots\ldots\ldots\ldots\ldots (37)$$

Table 4-4 Transit Inventory Cost

Processing Plants	Transit Inventory
Babol	20
Esfahan	23
Karaj	40
Kashan	10
Total	**93**
Transit Inventory cost	**1157.24**

Table 4-5 Recording Transit Inventory for a processing plant

	LPG Daily Deliveries to Babol Processing Plant					
	Days	Time periods (minutes)	Daily Demand (1)	Cummulative Demad	Daily Deliveries (2)	Transit Inventory (Tonnes) (1-2)
	0	0	0	0	0	0
	1	1440	45.92	45.92	2	9.92
	2	2880	45.92	91.84	1	27.92
	3	4320	45.92	137.76	2	9.92
	4	5760	45.92	183.68	1	27.92
	5	7200	45.92	229.6	3	-8.08
	6	8640	45.92	275.52	2	9.92
	7	10080	45.92	321.44	1	27.92
	8	11520	45.92	367.36	0	45.92
	9	12960	45.92	413.28	3	-8.08
	10	14400	45.92	459.2	2	9.92
	11	15840	45.92	505.12	0	45.92
	12	17280	45.92	551.04	3	-8.08
	13	18720	45.92	596.96	1	27.92
	14	20160	45.92	642.88	2	9.92
	15	21600	45.92	688.8	1	27.92
	16	23040	45.92	734.72	1	27.92
	17	24480	45.92	780.64	2	9.92
	18	25920	45.92	826.56	3	-8.08
	19	27360	45.92	872.48	4	-26.08
	20	28800	45.92	918.4	5	-44.08
	21	30240	45.92	964.32	1	27.92
	22	31680	45.92	1010.24	2	9.92
	23	33120	45.92	1056.16	0	45.92
	24	34560	45.92	1102.08	0	45.92
Time Horizon	25	36000	45.92	1148	2	9.92

4.7.2 Environmental cost

Transportation has a direct impact on environment. This impact could be assessed in three categories as follows:

- **Air Pollution:** Air pollution remains among the most recognised and widely accepted negative impact of the transportation system. To measure the impact of transportation schedule on the air pollution, the STEM (Surface Transportation Economic Analysis Model) Model [114] was used. This model uses emission modelling to forecast the quantity of emission resulting from each transportation alternatives for key pollutants. Tones of emitted pollutants are the output of this modelling exercise, to which a dollar value can be assigned, the STEM model includes per tone cost estimates for key pollutants, these cost estimates are as follows:

Pollutant	Cost per ton
Carbon Monoxide (CO)	$ 3,889
Hydrocarbons(HC)	$ 1,779
Nitrogen Oxides(NOx)	$ 3,731

Figure 4-54 Cost per ton of pollutant

Also [115] reports that the cost of avoiding CO_2 emissions is about US$30-50 per tonne of Co2 for most of the current range of fuel prices. To estimate the impact of a transportation schedule, the emitted pollutant from the transportation must be calculated. To this effect the daily fuel consumption and the average fuel economy for the available trucks in a schedule are calculated as follows. These formulations are based on work carried out in Texas Transportation Institute [116].

$$\text{Daily Fuel Consumption} = \frac{\text{Hours Travelled} \times \text{Average Speed}}{\text{Average Fuel Economy}} \quad \ldots\ldots\ldots\ldots\ldots\ldots (38)$$

$$\text{Average Fuel Economy} = 3.74 + (0.11 \times \text{Average Speed}) \times (\text{Liters per Kilometer}) \ldots\ldots\ldots\ldots\ldots (39)$$

The average speed and litters per Kilometre are provided by the logistics manager based on the historical data available. Based on the above formulation the total fuel consumption is calculated for the time horizon for all the trucks available in the schedule. The obtained fuel consumption is proportionate to the main pollutant and the total air pollution cost is obtained based on the specified cost units for each of these pollutants.

- **Water Pollution:** Transportation contributes significantly to water pollution and hydrologic problems. [117] suggests that 46 % of the vehicles in US leak hazardous fluids, including oil, transmission, hydraulic and break fluid and antifreeze. These pollutants are either burnt by the car's engine or these are improperly disposed into the ground or into sewers. The run-offs from streets, highways carry pollutants from the road surface into streams, rivers and lakes. These pollutants contaminate the dinking water increased flooding and leads to an increased need for flood control measures. The WSDOT (Washington Department Of Transportation) [117] estimates that the capital and operating costs associated with meeting run-off water quality and flood control requirements are about $0.002 to $0.005 per vehicle mile travelled. Using this figure the impact of a transportation schedule to the water pollution is calculated.
- **Noise Pollution:** Since the transportation is a major source of noise, the primary cost of noise is usually estimated based on the reduction of residential property value due to the traffic noise. [118] estimated the minimum cost of noise associated with automobile at $0.002 per vehicle mile travelled, and the maximum at $0.06 per vehicle mile travelled. This study used these figures to estimate the impact of a transportation schedule on the noise pollution.

4.7.3 Inventory Cost

Inventory carrying costs at Bottling plants result from storing and holding LPG for period of time and it is roughly proportional to the average volume of LPG kept in the reservoirs. In general the inventory cost can be considered in inventory holding costs and shortage costs. The inventory holding costs considers the following major elements:

- **Capital Costs**: this cost refers to the cost of the money tied up in the inventory. This cost may represent over 80 % of total cost.
- **LPG Reservoirs Service Costs:** Insurances and taxes are considered as part of the inventory holding costs because their level roughly depends on the amount of inventory available.

The out of stock inventory costs are incurred when a demand for LPG cylinders cannot be filled from the inventory. In the considering firm the inventory shortage cost is calculated based on the lost of sales. This occurs when the customers are faced with stock out situation and they may wish to withdraw their request for the LPG. Therefore, the cost is the profit that would have been made by this sale. Also, this could have additional cost for the negative effect that the shortage may have on the future sales.

In this application the inventory cost is calculated for each Bottling Plant existing in the system. The holding and shortage costs are specified for each respective plant in the system. The daily LPG deliveries to each Bottling plant are recoded. Based on the daily demands for the LPG and the initial inventory levels in each Bottling plants, the inventory cost is calculated. The inventory cost which is the summation of holding and shortage costs are obtained on daily basis. Table 4-6 demonstrates this procedure further.

Table 4-6 Partial demonstrations to calculate inventory costs for bottling plants

Input Data	Quantity	Units
Total Monthly Demand	1148	Tonnes
Time Horizon	25	Days
Available Inventory	180	Tonnes
Daily demand	45.92	Tonnes
Inventory Holding Cost	2	Euros
Inventory Shortage Cost	4	Euros

Period(t)	1	2	3	4	5	6	7	8	9	10
Initial Inventory	180	134.08	88.16	96.24	158.32	112.4	66.48	20.56	-25.36	-71.28
Demand(t)	45.92	45.92	45.92	45.92	45.92	45.92	45.92	45.92	45.92	45.92
Delivery Quantity(t)	0	0	54	108	0	0	0	0	0	0
Inventory Level (t)	**134.08**	**88.16**	**96.24**	**158.32**	**112.4**	**66.48**	**20.56**	**-25.36**	**-71.28**	**-117.2**
Holding Cost(t)	268.16	176.32	192.48	316.64	224.8	132.96	41.12	0	0	0
Shortage Cost (t)	0	0	0	0	0	0	0	-101.44	-285.12	-468.8
Total Inventory Cost(t)	**268.16**	**176.32**	**192.48**	**316.64**	**224.8**	**132.96**	**41.12**	**101.44**	**285.12**	**468.8**

4.7.4 Safety Cost

The safety cost is the average monthly cost due to possible fatal, injury and property damages. Typically, the cost of vehicle accidents can be quite high, particularly in case of butane gas, which is highly explosive. This cost includes vehicle and other property damage as well as personal injury and mortality. For the most part of the accidents drivers are covered through personal auto and life insurance policies, which is considered in the ownership cost. However, some accident costs are not covered by insurance policies. Insurance deductibles in this category are damages and losses beyond the measure of affordable coverage.

Table 4-7 identifies the categories for these costs and it provides some estimate cost for each of these categories [114].

Table 4-7 Safety cost breakdown

Accident Type	Internal Cost	External Cost
Fatal accident	\$ 2,317,398	\$408,952
Injury accident	\$50,760	\$8,958
Property damage	\$2,824	\$498

To calculate the safety cost, the accident rate (α) for each transportation schedule is calculated as follows [114]:

$$\alpha = \frac{\sum_{r=1}^{n} \text{Total Number of Accident}_r}{\sum_{t=1}^{m} \text{Distance Travelled}_t + \sum_{t=1}^{m} \text{Working Hours}_t} \quad \text{.......................} (40)$$

$$\textit{Where}$$
$$r = 1,...n\ (r = route)$$
$$t = 1,...m\ (t = trucks)$$

The total number of accident is obtained per route considered in the system. This is calculated using the following equation [114].

Total Number of Accidents $_r$= Accident Frequency $_r$ * Route usage .... (41)

The accident frequency, is the number of accidents occurred in a specific transportation route. This is identified for each route based on historical safety data. Also, the route usage is the total number of times that a specific route is used in a transportation schedule. The safety cost for a schedule (SC) is calculated as follows:

$$SC = \alpha \times \text{Company's Safety Cost} \quad \text{..........................} (42)$$

4.7.5 Service Level Cost

The service level is a metric used to measure the percentage of daily demands satisfied. This measure is calculated using two approaches as follows:

- **Practical approach:**

Based on the considered time horizon in a transportation schedule, the daily-required demand for each bottling plant in the system is established. For each time period in the time horizon, the number of deliveries to the bolting plants is recorded. The daily service level for each bottling plant in the system is calculated using the following formula:

$$\begin{gathered}\text{if Delivered Quantity}_t > 0,\\ \text{then}\\ SL_t = \frac{\text{Numberof Deliveries}_t}{\text{Daily Demand}_t} \times 100\\ \text{else}\\ SL_t = 0\end{gathered} \qquad \text{.......................... (43)}$$

The daily service level is reordered for each bottling plant and at the end of the time horizon the average satisfied daily service level is obtained. If the average service level does not reach the average targeted service level defined for each bottling plant then there is an association service level cost. The service level cost is related to the revenue cost provided by the company. The following formulates how this cost is calculated.

$$\begin{gathered}\textit{Unsatisfied Service Level} = \textit{Average Service Level Target} - \textit{Average Service Level}\\ \textit{Service Level Cost} = \textit{Unsatisfied Service Level} * \textit{Revenue Cost}\end{gathered} \qquad \text{.......................... (44)}$$

- **Theoretical approach:**

In this approach the service level for each bottling plant is obtained as above. When the service level is less than the targeted level then the following profit formula is used to calculate this cost as follows [1]:

$$\begin{gathered}P = \alpha \times SL^{1/2} - \beta \times SL^2\\ \text{where}\\ \alpha = \text{Revenue Cost Slope}\\ \beta = \text{Logistic Cost Slope}\end{gathered} \qquad \text{.......................... (45)}$$

In this approach, the profit related to 100% service level is obtained. Then the profit related to the corresponding service level is calculated using the above formulation. The subtraction of profit related to the service level form the 100% service level profit is the corresponding service level cost.

4.7.6 Transportation Cost Formulation

The transportation cost related to a delivery schedule could be formulated according to any of the following categories.

- **Uniform rate Transportation cost:** This is the transportation rate specified by the logistics manager. This rate is specified for both the loaded and unloaded distances that a truck has travelled. The uniform rate is considered to be the same for all the routes. This rate does not consider any specific characteristics of the route taken for transporting the required volume of LPG. This cost depends on the volume shipped and the distance between plant and warehouse. In this calculation the uniform rate is established for transporting 18 tonnes of LPG. The 18 tonnes is considered as the unit volume for transportation. Total transportation cost is the sum of all transportation cost for each individual truck considered in a schedule. The overall calculation for this rate is as follows:

  Total Transportation cost = (Loaded Transportation Distance × Uniform Rate) + (Unloaded Transportation Distance × Uniform Rate)

  $$\sum_{T=1}^{m}\sum_{LR=1}^{n}(d_{Lr} * \text{UniformRate}) + \sum_{T=1}^{m}\sum_{UR=1}^{n}(d_{Ur} * \text{UniformRate}) \quad \text{..................... (46)}$$

- **Proportional Transportation cost:** This rate is different for each route considered in transportation scheduling. This rate is based on the travel time taken in each route. Travel time refers to the value of time spent on transportation, including time devoted to waiting, accessing vehicles, congestion time and also cognitive time. The cognitive time, incorporate various qualities of service attributes such as comfort and safety.

  The travel time varies and depending on factors such as driver comfort, road conditions, congestion rate. The driver comfort depends on the availability of facilities along the considering route. Road conditions refer to the type route considering such as express way, high way, and non express roads also other adverse geographical conditions that would affect the selection of the road for transporting LPG. Further more congestion rate refers to possible traffic and delays that may occur during the transportation. This factor reflects the peak, off peak, night or day time transportation.

The proportionate cost factor is specified considering both loaded and unloaded trucks. For each truck in the system the routes taken by that truck is obtained and the corresponding proportionate cost is calculated for each truck.

Total Transportation cost = (Loaded Distance travelled* proportionate loaded Rate +unloaded distance travelled* proportionate unloaded Rate)

$$\sum_{T=1}^{m}\sum_{LR=1}^{n}(d_{Lr} * \text{Proportionate Rate}_{\text{Lr}}) + \sum_{T=1}^{m}\sum_{UR=1}^{n}(d_{Ur} * \text{Proportionate Rate}_{\text{Ur}}) \qquad (47)$$

- **Tapering Transportation Cost:** The following exponential function is used to obtain the transportation cost.

$$Y = aX^{b} \qquad (48)$$

Y is the transportation cost, ***X*** is the distance travelled, ***a*** is the slop value which represents the total fixed cost related to the delivery schedule and ***b*** is intensity. The intensity could reflect the convenience factors in using a route in the schedule. Using this function both loaded and unloaded transportation cost is obtained.

4.7.7 Truck Costs:

In this work, trucks cost as the main resource in LPG transportation is considered. Trucks are either leased or owned by the company. If a truck was leased then the only cost would be the leasing charges per truck that the company needs to pay. However, if companies owned vehicles are used then a transportation service incurs a number of costs. These costs can be categorised as either fixed or variable costs.

Fixed costs are those costs that are constant over the normal operating period of the carrier and all other costs are treated as variable. For instance, fixed costs are those associated to roadway acquisition and maintenance, depot facilities, transport equipment, and administration related costs. Variable costs usually include costs such as fuel, labour, equipment maintenance, handling, and pickup and delivery. Costs related to trucks are considered in two categories, namely trucks' operation and ownership costs as shown in

Table 4-8.

- **Vehicle ownership costs:** the vehicle depreciation cost is a function of both time and usage. The vehicle ownership and depreciation can be estimated as a cost per vehicle mile travelled. Also, other cost factors as part of this category are such as insurance and financing, registrations and road tax. In this approach these cost are estimated and converted to a vehicle mile equivalent.
- **Operational costs*:*** the vehicle variable costs are mainly considered here such as fuel, maintenance tires and oil. These cost factors are also calculated based on vehicle mile equivalent.

Table 4-8 associated truck costs

	Category	Supper Trucks	Mid Sized Trucks	Small Sized Trucks
Operation costs (Cents/Miles)	Gasoline & Oil	6.5	5.7	4.8
	Maintenance	3.7	3.4	3.1
	Tires	1.4	1.6	9.2
	Sub Total	**11.6**	**10.7**	**17.1**
Ownership Costs (Cents/Miles)	License & Registrations	1316	885	1012
	Finance Charges	410	223	175
	Depreciation	3648	3355	2871
	Road tax	958	812	603
	Subtotal	**6332**	**5275**	**4661**

4.7.8 Human Factor Cost

Industrial actions would often affect transportation operations. This is usually caused by dispute involving handling route assignments to the drivers. An effective route assignment mechanism could provide a solution preventing any possible delays in delivering raw materials to bottling plants.

In general selected routes may be disliked by the drivers due to many reasons such as inconvenience, long distances, adverse geographical conditions, lack of transit facilities and etc. To incorporate these factors in route selection, **Human factor** factor is introduced for each route in the system. This factor is used as an upper limit to suggest the total number of times that this route may be used in a transportation schedule. Also, this factor could be used to determine the competitive and none competitive routes. Considering the competitor's activities in different local markets the company can introduce incentives for more activities in different zones. Therefore, this factor could be used to relax or further constraint selection of particular routes leading to a particular market of interest. Table 4-9, illustrates how human factor cost is typically calculated for a transportation schedule.

Table 4-9 Human Factor Cost calculations

Union Cost (a)	0.17
Union Intensity (b)	2

Routes	Distance	Union Factor (1)	No. times used in the schedule (2)	(1-2)<0 (X)	$Y = aX^b$
1	1300	10	8	2	0.00
2	500	20	25	-5	4.25
3	1000	30	30	0	0.00
4	1250	10	10	0	0.00
5	1900	10	10	0	0.00
6	2000	20	25	-5	4.25
7	750	20	26	-6	6.12
8	650	30	25	5	0.00
9	120	10	15	-5	4.25
10	145	10	10	0	0.00
			Total Union Cost		18.87

As shown above, the human factor is obtained per route based on the exponential function. In this function (X) represents the deviation from human factor per route. Also, factor (a) represents the human factor cost per violation and (b) is the intensity factor representing the importance of the human factor route limit.

4.7.9 Work balance cost

A transportation schedule is typically composed of a number of trucks travelling a number of routes. In a schedule a resource may be under/over utilised based on the route assignments. To balance the workload amongst the resources fairly and evenly the work balance cost parameter was used.

This cost parameter is used to measure the resources ideal time and the possible over time. In this measure, time gap for each truck is calculated. The time gap is the difference between the trucks available time and the trucks busy time (i.e. travelling time). The available time is established based on the standard daily working hours and also this metric includes the expected down time duration of the truck. Usually trucks are expected to be available 70% of the time in a week period.

A schedule ideal and over time is calculated using the mean squared idle time for the available resources in the schedule. The following formula is used to calculate the mean squared idle time for the schedule.

$$Z_{idel} = \sum_{t=1}^{m} \frac{(\text{Available time}_t - \text{Busy time}_t)^2}{m} \quad \text{......................... (49)}$$

where

m = number of trucks

After obtaining the mean squared idle time for a schedule this time is multiplied by the capital cost per time unit. The capital cost here is the summation of labour and truck cost per hour. These costs factors are provided by the logistics management.

4.8 Mathematical formulation

The considering problem can be categorised as type of classical transportation problem, originally developed by F.L. Hitchcock [101]. This is one of the combinatorial problems involving constraints that have been studied. In most cases, it is required to solve the problem considering more than one decision criterion and therefore giving place to the multi-objective transportation problem. In general the considering problem can be described as transportation of LPG as a homogenous product from m sources to n destinations. The sources are LPG production facilities, refineries or supply points, characterised by available capacities a_i, for i =1,..., m. the destinations are consumption facilities, which are bottling plants, characterised by required levels of demand b_j, for j = 1, 2,..., n. considering e_k (k =1, 2,..., K) to be the units of LPG product which can be carried by truck with K different capacities.

A penalty C_{ijk}^{P} is associated with transportation of a unit of LPG from source i to destination j by means of the *k-th* conveyance for the *p-th* decision criterion. In this work, the decision criterion relates to factors such as transportation cost, delivery time, quantity of goods delivered and etc.

The aim is to determine the amount of the product x_{ijk} to be transported from all sources i to all destinations j by means of tucks with k capacity so that all P decision criteria are taken into account in a way satisfactory to decision maker.

In this application, the transportation problem is a multiple objectives of non-linear type. The non-linearity could be considered both in objective functions and constrains. As there are measures, which are not proportional to the amount of LPG transported. For example applying tapering cost measure to obtain transportation cost represents no proportional relationship between the cost of the route and the amount transported. This is multi-objective non-linear transportation problem, which can be mathematically formulated as follows:

$$Minimise\ Z_p = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{k} C_{ijk}^{P} X_{ijk},\quad P=1,2..,P \qquad \text{.................. (50)}$$

The subscript on Z_p and C_{ijk}^{P} denote the *Pth* penalty criterion. In this approach, all the objectives are combined into one overall cost function and then the new formulation of the problem becomes the minimisation of the main cost function by satisfying all the existing constraint as in the system. The cost for a considering factor also reflects its importance in relation to the other parameters considered in the objective function formulation. The following illustrates the mathematical formulation of the cost objects.

1. **Transportation Cost (TC):** This criterion is related to loaded (*LTC*) and unloaded (*ULTC*) transportation cost. The aim is to minimise these cost factors in generated schedules. Transportation costs are formulated as follows:

$$Z_{LTC} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{k} C_{ijk}^{LTC}\ X_{ijk} \qquad \text{.................. (51)}$$

$$Z_{ULTC} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{k} C_{ijk}^{ULTC}\ X_{ijk} \qquad \text{.................. (52)}$$

2. **Truck Cost (TKC):** This criterion is to find the proper fleet size for the transferring quantity X_{ijk}. This can be formulated as:

$$Z_{TKC} = \sum_{k=1}^{k}\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ijk}^{TCK} X_{ijk} \qquad \text{............... (53)}$$

Where *K* is the truck used, and *TCK* is related to individual truck cost, which is based on the operation and ownership costs for the truck considered.

3. **Inventory cost (INVC):** This criterion is monitored on daily bases. The aim is to find the holding (*HC*) and shortage costs (*SC*) for each bottling plant (b_j) over the planning time horizon. This objective can be formulated as follows:

$$Z_{HC} = \sum_{TH=1}^{TH}\sum_{i=1}^{m}\sum_{j=1}^{n} C_{j}^{HC} X_{ijTH} \qquad \text{............... (54)}$$

$$Z_{SC} = \sum_{TH=1}^{TH} \sum_{i=1}^{m} \sum_{j=1}^{n} C_j^{SC} X_{ijTH} \qquad \text{..............(55)}$$

Where, subscript *TH* represents time period (*t*) in the specified Time Horizon. Also *HC* and *SC* represent holding and shortage inventory costs respectively. The total inventory cost for the system is the summation of these cost factors.

4. **Truck Work balance:** This criterion aims to minimise the resource ideal time used within the schedule.

$$TravelTime_k = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ijk}^{TT} X_{ijk} \qquad \text{..............(56)}$$

$$Z_{idel} = \sum_{k=1}^{k} \frac{(Availabletime_k - TravelTime_k)^2}{k} \qquad \text{..............(57)}$$

Where C_{ij}^{TT} represents the travel time from source *i* to destination *j*. Also the ***Available time*** is the total expected truck available time based on trucks schedule and unscheduled maintenance events. This is a given value for each truck.

5. **Service Level (SL):** This is used to maximise the service level provided by the system. This is achieved by minimising the mean unsatisfied service level.

$$Z_{SL} = \sum_{i=1}^{m} \frac{Max\,(0, (Target\ Service\ level_i - Actual\ Service\ Level_i))}{m} \qquad \text{...........(58)}$$

6. **Safety Cost (SAFC):** This criterion is used to minimise total possible accident rates for established schedules.

$$Z_{SAFC} = \frac{\sum_{k=1}^{k} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ijk}^{Acc} X_{ijk}}{\sum_{k=1}^{k} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ijk}^{Dist} X_{ijk} + \sum_{k=1}^{k} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ijk}^{TT} X_{ijk}} \qquad \text{...........(59)}$$

Where, C_{ijk}^{Acc} represents the Accident frequency, C_{ijk}^{Dist} distance and C_{ijk}^{TT} travel time for each route.

7. **Human factor Costs (HFC):** This is used to minimise the use of in appropriate routes in a schedule. Therefore this objective aims to minimise deviation from human factor factor defined for each considering route in a schedule.

$$Z_{HFC} = \sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n}(X_{ijk} - HumanFactorLimit_{ij}) \qquad \text{..........} (60)$$

8. **Schedule Completion Time (SCT):** The completion time is calculated for each available truck in the schedule (***k***) and the maximum of all the completion time is the total schedule completion time.

$$\text{Trave lTime}_k = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ijk}^{TT} X_{ijk} \qquad \text{..........} (61)$$

$$Z_{SCT} = Max(Trave\, lTime_k) \qquad \text{..........} (62)$$

In the above formulation C_{ijk}^{TT} indicates the travel time for each route taken by truck (***k***) and X_{ijk} represents the route taken by truck (***k***).

9. **Environmental Cost (EC):** This criterion is based on minimising environmental pollution based on the transportation schedule. The following demonstrates the general environmental factors considered in this formulation.
   - Air pollution (ECA):

$$Z_{ECA} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{k} C_{ijk}^{ECA} X_{ijk} \qquad \text{..........} (63)$$

   - Water Pollution (ECW):

$$Z_{ECW} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{k} C_{ijk}^{ECW} X_{ijk} \qquad \text{..........} (64)$$

   - Noise Pollution (ECN):

$$Z_{ECN} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{k} C_{ijk}^{ECN} X_{ijk} \qquad \text{..........} (65)$$

4.9 Constraints

Constraints are usually classified as equality or inequality relationships. As the equality may be assumed into the system, therefore the inequality relationships are important to be considered. In this application constraints are considered as hard and soft constraints. In general hard constraints refers to those constraints that cannot be violated by any means. Violation of such constraints results in infeasibility of the found solution. In the other hand soft constraint refers to those limits that can be violated and does not impact the feasibility of the solution. Penalty methods are used to justify the violation of the soft constraints.

Typically, the generated solutions (i.e. schedules) are evaluated using the objective function, and the feasibility of the solution is confirmed by checking if any hard constraints are violated. If not the schedule is assigned the fitness value corresponding to the objective function evaluation. If any hard constraints are violated, the solution is infeasible and thus has no fitness. In such case the solution are repaired using the repair techniques developed for this application.

However, in many practical applications such as the current field of investigation are highly constrained, therefore finding a feasible point is almost as difficult as finding the best solution. As a result, it is usually desired to get some information out of infeasible solutions, perhaps by degrading their fitness ranking in relation to the degree of constrain violation. This is achieved through penalty method. The penalty method applies to the soft constraints here in this application. In the penalty method, the constraint problem in optimisation is transformed to an unconstrained problem by associating a cost or penalty with all constraint violations. This cost is included in the objective function evaluation. Therefore considering the penalty method the general transportation problem can be formulated as in the following equation. Where, λ_p represents the penalty cost for violating constraints on each objective.

$$Minimise\ \ \mathrm{Z_p} = \sum_{p=1}^{P}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K} C_{ijk}^{p}(X_{ijk}) + \lambda_p \qquad \text{.................. (66)}$$

The following sections aim to describe any hard or soft constraints considered here in this application.

4.9.1 Completion time cost

- **Hard Constraint:** In this considering system, schedules are to be completed on monthly bases. The user can specify the time horizon to establish the schedule within this limit. The time horizon is considered as a hard constraint, which cannot be violated. Therefore, schedule completion time must not exceed the specified time horizon.

$$Max(Trave\,lTime_k) \le Time\ Horizon \qquad \text{.................. (67)}$$

Where $k = \{1,..., k\}, numberoftrucks$

- **Soft Constraints:** this allows further restrictions on the completion time. In this approach the user is allowed to specify a range of time windows or limits that completion time should stay in. When, these time limits are violated the generated solutions are not infeasible. In fact in such cases penalties are assigned to increase the cost of the schedule. In this application the user is prompted to specify time limits and also penalty cost in case of constraint violation. Table 4-10, demonstrates an example for such time limits and also penalty cost considered at each level. Here, ***Comp*** represents the calculated completion time and penalty costs are based on overtime-labour charges. Figure 4-55 illustrates the specified limits graphically.

Table 4-10 constraints on completion time

Completion time	Penalty
$0 < Comp \leq 24$	0
$24 < Comp \leq 25$	36486
$25 < Comp \leq 26$	72971
$26 < Comp \leq 27$	109457
$27 < Comp \leq 28$	145942
$29 < Comp \leq 30$	182428
$30 < Comp \leq 31$	218914

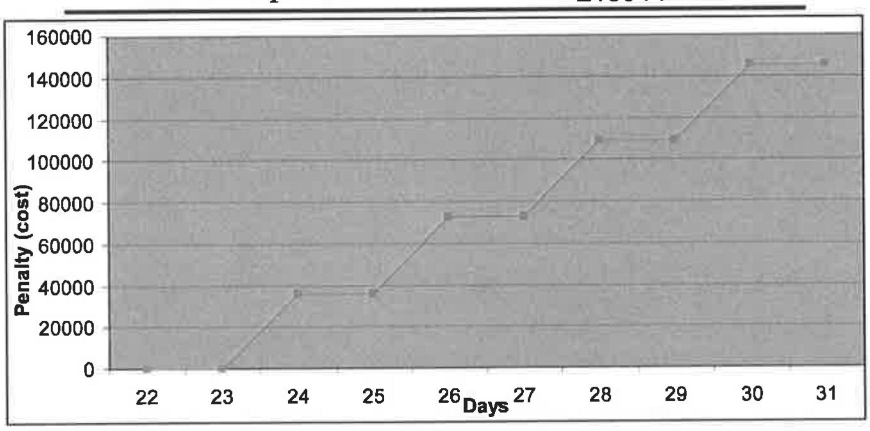


Figure 4-55 Penalty cost on violating the completion time constraints

4.9.2 Environmental cost

- **Hard Constraint:** the user has the option to set lower and upper limits on environmental cost. However, here the aim is to minimise this cost measure as much as possible. For this purpose, soft constraint approach is considered here.
- **Soft Constraint:** the user can specify different cost limits that the environmental cost should stay within these limits. Upon the violation of such limits penalties are assigned according to the identified cost parameters. Table 4-11, shows an example of soft

constraints on the environmental cost. Also, Figure 4-56 illustrates the non-linear penalties assigned to schedule cost when these limits are violated.

Table 4-11 Constraints on Environmental costs

Environmental cost (Env)	Penalty
$0 < Env \leq 500$	15.81
$500 < Env \leq 1000$	38.73
$1000 < Env \leq 2000$	59.16
$2000 < Env \leq 5000$	86.60
$5000 \leq Env$	111.60

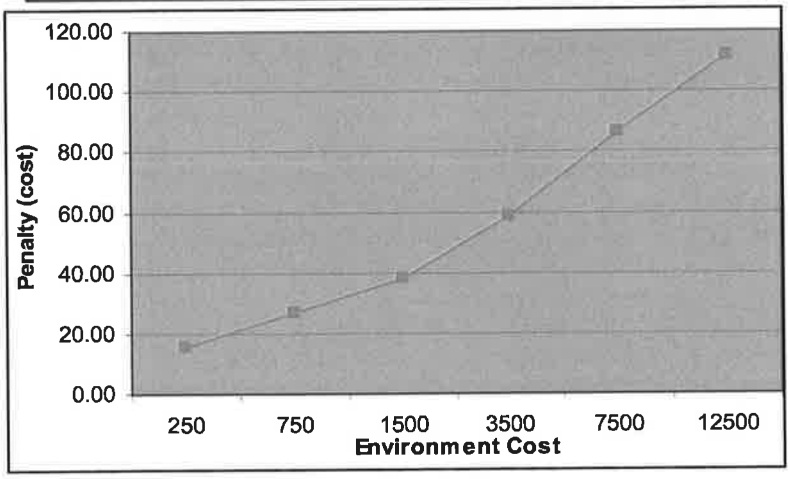


Figure 4-56 non-linear penalties for environment cost

4.9.3 Inventory Cost

- **Hard Constraints:** There are two general conditions to consider in controlling the inventory costs.
    - **Demand = supplied quantity:** this suggests that total quantity transported to processing plants is equal to the demand rate at the processing plant.
    - **Demand <> supplied quantity:** This suggests that the total quantity transported to processing plants could be more or less than the required demands.

The user can choose any of the above options to set the control limits on the inventory levels. Any violations of these constraints make the solution infeasible and as a result the solution must be disregarded or repaired to meet the specified inventory condition.

- **Soft Constraints:** the above conditions can be modelled as soft constraints and therefore penalties are assigned when conditions are not met. In such cases schedules encounter higher costs. In this approach irrespective of the above set conditions, penalties are assigned when the demand level at processing plants are not met. If the demand at (bj) is not met, then penalty assigned for violating this constraint is calculated as follows:

$$\lambda_{Inv} = \begin{cases} (b_j - X_{ij}) * \text{Shortage cost}, & \text{if } X_{ijk} \le b_j, \\ (X_{ij} - b_j) * Holding \text{ cost}, & \text{if } X_{ijk} \ge b_j, \end{cases} \qquad \text{.............. (68)}$$

Where, X_{ij} is the amount of LPG transported from refineries (i) to bottling plants (j), b_j is the demand level at processing plant j.

4.9.4 Safety Cost

As described earlier, the safety cost is established based on the historical data available on the number of accidents on each considering route. In reality there should not be any tolerance for any accident in transporting LPG to processing plants. This is achieved in the following ways:

- **Hard Constraints:** The user can specify an upper limit for the safety cost that the company is budgeting to run the transportation of raw materials to processing plants. In this way schedules exceeding the specified safety cost are either rejected or repaired to meet the safety limit.

$$\sum_{j=1}^{n}\sum_{i=1}^{m} X_{ij} \le SafetyFactor, \qquad j = 1, 2, ..., \text{n}, \qquad \text{................ (69)}$$

- **Soft Constraints:** The user is prompted to specify safety cost limits and corresponding penalties for violating such limits. The safety cost here is related to fatal injury cost. Table 4-12 demonstrates some soft constrains on the number of accidents and the associated penalty costs. Figure 4-57 illustrates safety penalty costs graphically.

Table 4-12 Constraints on the number of accidents

No. Accidents (Acc)	Penalty
$0 < Acc \le 1$	639
$1 < Acc \le 2$	904
$2 < Acc \le 4$	1107
$4 < Acc \le 6$	1429
$6 < Acc \le 8$	1691
$8 < Acc \le 10$	1918

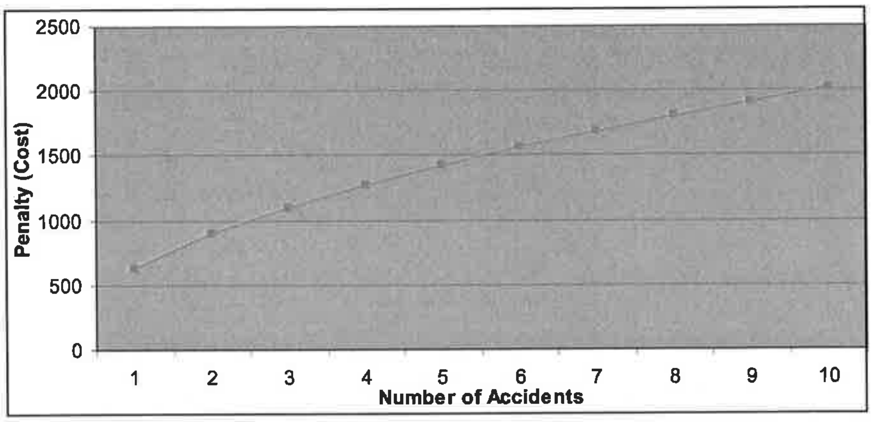


Figure 4-57 Penalties on the safety cost violations

4.9.5 Service Level Cost

- **Hard Constraints:** usually schedules with very low service levels are ignored. In this way the user can specify a lower limit on the acceptable service level. Schedules providing service levels less than the specified value is rejected or repaired to meet the target service level. Alternatively the user can use penalty methods to assign high weights to schedules with low service levels.
- **Soft Constraints:** in this approach the user can specify associated penalties for different service levels. Schedules having low service levels are penalised detrimentally. Table 4-13, illustrates a series of constraints on the service level and associated costs or penalties on these levels. Also, Figure 4-58 demonstrates, how the associated penalties are declined as the service level increased. The penalty cost is based on the revenue cost.

Service Level (%)	Penalty
$0 < SL \leq 10$	16404
$10 < SL \leq 20$	12729
$20 < SL \leq 30$	11267
$30 < SL \leq 40$	10295
$40 < SL \leq 50$	9577
$50 < SL \leq 60$	9011
$60 < SL \leq 70$	8544
$70 < SL \leq 80$	8148
$80 < SL \leq 90$	7805

Table 4-13 Constraints on Service Level

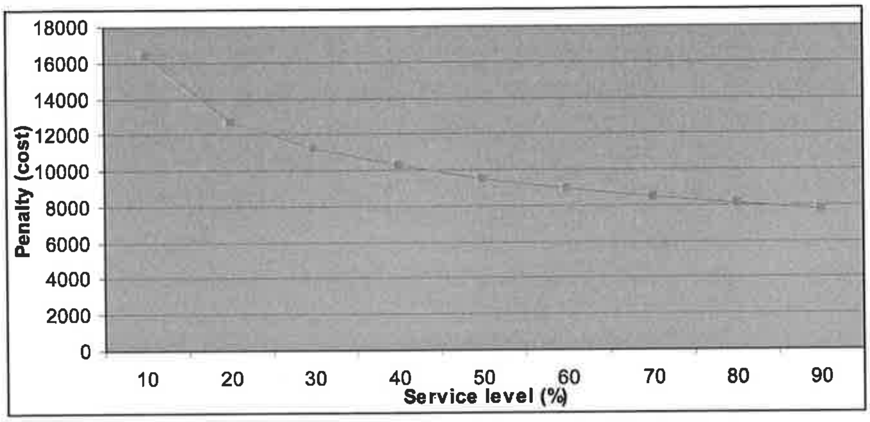


Figure 4-58 Penalties of service level violations

4.9.6 Transportation Cost Formulation

- **Hard Constraints:** There is an upper bound on the total distance travelled. The user based on the system requirements defines this upper bound. Here, this upper bound is set to 500,000 km. Schedules with greater value for this parameter is ignored.

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{k} X_{ijk} \leq 500{,}000 \text{ Km} \qquad \text{..................(69)}$$

- **Soft constraints:** There are no considerations for soft constraints on distance travelled. However, the user has the option to define further constraints and corresponding penalties on this parameter.

4.9.7 Truck Costs:

- **Hard Constraints:** Here the hard constraint is left on the total number of trucks to be used in a schedule. Here, there is a lower and upper limit set for the available number of trucks in the system. The user sets these values.

$$1 \leq \sum_{k=1}^{l}\sum_{i}^{m}\sum_{j}^{n} X_{ijk} \leq 40 \qquad \text{..................(70)}$$

- **Soft Constraints:** There are no considerations for the soft constraint here. However, the user can add further constraints on the number of trucks used in schedule.

4.9.8 Human Factor Cost

The followings are example of hard constraints considered for this model.

- **Upper Limit on the Human Factor:** This could be used as limiting the maximum number of times that a route can be used in a schedule. In this approach users can specify an upper limit on the total number of deviations from these factors. Schedules exceeding such limits are not anymore feasible and therefore they are to be repaired.

- **Route Combinations:** There are situations that specific routes are not to be assigned one after each other. For instance assigning two consecutive long trips to a driver is typically not acceptable by either drivers or unions. To this effect this application allows one to specify these routes. Once a schedule is generated, a check is done to make sure that the specified route combinations do not exists in the schedule. Existence of such combinations would lead to infeasibility of the schedule.
- **Route Matching:** In contrast to the above constraints, there may exist cases that unions may request to travel specific routes one after each other. For example, a very short tripe must be followed by a moderately long trip. This constraint is provided to allow the flexibility to model such events. Similar to the above constraint, schedules are check for matching the specified routes. If a schedule does not hold this matching conditions then this schedule is repaired.
- **Soft Constraints:** There are no considerations for the soft constraint here. However, the user can add further constraints on the human factor factor used for each route to limit route usage within a schedule.

4.9.9 Work balance cost

- **Hard Constraints:** The followings are a set of hard constraints considered for this measure.
- **Trucks Availability:** Considering down time and breakdown periods, trucks would be available for certain duration. The user specifies this availability value. For instance, considering a week period of 168 hrs, a truck would be available for 70% of this time, which is 117.6 hrs. Therefore a constraint is set to make sure that the total workloads assigned to trucks do not exceed this time limit.

$$\sum_{k=1}^{l}\sum_{i}^{m}\sum_{j}^{n} T_{ij} X_{ijk} \leq \text{Available Time}_k \qquad \text{.................. (71)}$$

- **Number of operations:** operations are deliveries to bottling plants. To balance workload across resources, numbers of deliveries to bottling plants are to be balanced between resources. Considering the total number of resources used in a schedule and the total LPG capacity to be distributed, the possible number of deliveries to be taken by each resource is specified. Adhering to these limits would prevent assigning too much workload to one resource.

$$\sum_{k=1}^{l}\sum_{i}^{m}\sum_{j}^{n} X_{ijk} \leq \frac{1}{\text{Total Number of Trucks}} \% \times \text{Total number of Delivery operations}$$

$$\text{Where K} = 1,\ldots,\text{L}.$$

.... (72)

$\boldsymbol{K}$ represents trucks in the model and X_{ijk} is used to obtain the total number of deliveries made by truck $\boldsymbol{K}$. In addition, the user can specify minimum number of delivery operations that a truck in a schedule must undertake. This is shown as follows:

$$\sum_{k=1}^{l}\sum_{i}^{m}\sum_{j}^{n} X_{ijk} \geq 10$$

.... (73)

$$\text{Where K} = 1,\ldots,\text{L}.$$

- **Soft Constraints:** In this approach, penalty cost is assigned to trucks' ideal times. In this way, once a schedule is generated, the ideal time for each resource (i.e. truck) is obtained. These ideal times are summed and the following non-linear approach was conducted to assign penalties based on the obtained value. Table 4-14, illustrates a series of constraints on the trucks' ideal times and associated costs or penalties on these values. These penalties are based on ideal time capital cost. Also, Figure 4-59, demonstrates, how the associated penalties increase as the ideal time deviates from 60 minutes.

Table 4-14 Constraints on Ideal time

Service Level (%)	Penalty
$0 < IdealTime \leq 60$	0
$60 < IdealTime \leq 360$	194
$360 < IdealTime \leq 660$	274
$660 < IdealTime \leq 960$	336
$960 < IdealTime \leq 1260$	388
$1260 < IdealTime \leq 1560$	434
$1560 < IdealTime \leq 1860$	475
$1860 < IdealTime \leq 2160$	513
$2160 < IdealTime$	549

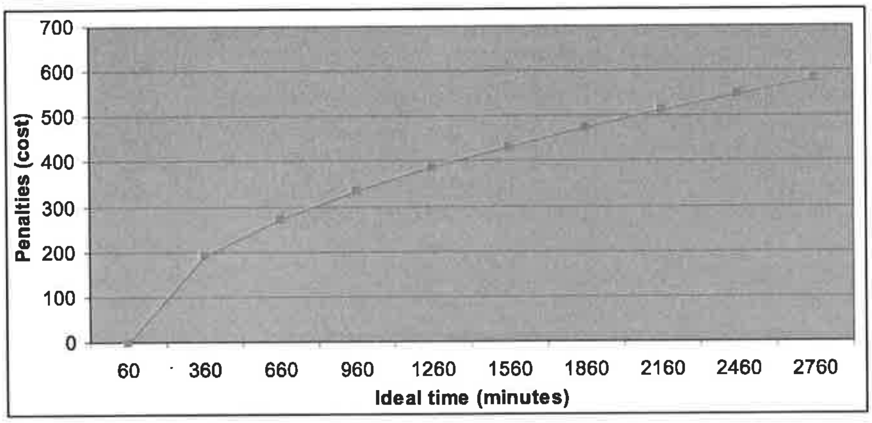


Figure 4-59 Penalties on Ideal time

4.10 Conclusions

The purpose of this chapter was to demonstrate the design concepts and the development of the Search Engine module. The related issues in designing this module including the basic architectural requirements, the main design components of the search engine and their functionalities were described in detail. Also in this regard, the process involved and functions developed to facilitate Genetic algorithm and Pareto Optimal search were described in detail.

In addition, different cost parameters reflecting the main considering objectives in managing the logistics activities were presented. To this effect the objective function formulation used to evaluate the fitness of the generated schedules was described. Also, the considering system constraints impacting the feasibility of the generated solutions were explained.

Using this application the constraints could be modelled as either hard or soft constraints. The hard constraints impact the feasibility of the generated solutions. Upon the violations of such constraints the repair techniques are used to address the feasibility of the generated solutions.

The soft constraints do not violate the feasibility of the generated solutions. The penalty method was considered to allow soft constraint violation. The penalty method adds further cost to the schedule cost generated. In this way the solution space is extended to include the relative non-feasible regions into the solution space. Some of the main points drawn from this chapter are as follows:

- The chromosomes are encoded to represent a complete monthly schedule for transporting LPG from refineries to processing plants in different localities.
- The main parameters considered in encoding chromosomes are such as trucks, routes and start times. These decision variables are randomly chosen from their respective decision spaces. This method of coding is known as phenotypic. Some of the components of genetic search algorithms developed here are:
  - ◊ The fitness of the chromosomes in this application is based on nine different objective functions.
  - ◊ The objective functions considered here are subject to real system constraints.
  - ◊ This application was designed to handle linear and non-linearity's relationship exists within both objective functions and constraints.
  - ◊ Three scaling methods were introduced to prevent any possible premature convergence. These are such as 0-1 scaling, linear scaling and ranked linear scaling.
  - ◊ The selection method developed is based on roulette wheel and tournament selection principles.
  - ◊ The main crossover techniques developed here are based on single point fixed length and variable length methods and also partially mapped uniform crossover method.
  - ◊ The mutation methods are based on classic and inversion principles.
  - ◊ Finally, the replacement methodology used is based on Elitism concepts.

The developed search engine incorporated unique features providing a better flexibility in modelling logistics systems. It extensively works in combination with Microsoft SQL server database, to transfer, record and retrieve data. This module could be used as either a stand-alone tool or in conjunction with simulation applications. It allows multi-criteria optimisation. Also, it considers linear and non-linear relationships in terms of system objectives and constraints. It further provides incorporation of different constraints from the real system. This application facilitates a better approach on route selection, environment al factors, safety, inventory level, and human factor consideration, which are typically neglected in VRSP decision-makings. However, further dynamic, stochastic behaviours and other complex relationship exist in the system are captured using simulation modelling techniques. The simulation module is the second component of the proposed DSS. The following chapter intends to establish an understanding of this technique and how it could be used for this propose.

Chapter 5: Simulation Modelling

5.1 Introduction

Based on the architectural design of the decision support system presented in chapter four, the search engine composed of genetic algorithm, which is used to provide a set of optimal transportation schedules. These schedules are generated having satisfied the objective functions and constraints considered for the system. The second module of the proposed architecture is a simulation engine. In general simulation model is used to captures dynamic and stochastic behaviours of the system, the complex relationship between supply sources, transportation and demand centres. It also allows modelling uncertainties resulting from resource availability, inventory levels and customer demand patterns. The aim of this module is to evaluate the generated optimum schedules from the Genetic search engine before implementing any schedule. In this way, this module is to maintain better credibility and confidence on generated schedules and also provide valuable insight into the VRS problem.

The primary aim of this chapter is to introduce the concept of simulation modelling and it advantages and disadvantages. Furthermore, the particular application of simulation modelling within logistics and supply chain management are studied. In addition the application of simulation and optimisation are reviewed. This chapter is continued on representing simulation tool and their classifications, software selection features and in particular features for witness simulation tool is described. Finally, alternative approaches to simulation modelling are presented.

5.2 Computer Simulation

Simulation is an indispensable problem solving methodology, which has been applied to a wide range of disciplines assisting decision making in planning, design and control of complex systems. [120] indicates that simulation involves the modelling of a process or system in such a way that the model mimics the response of the actual system to events that take place over time. In its broadest sense, simulation model involves process of designing a model of a real system and conducting experiments with this model for the purpose of understanding various strategies for the operation of the system.

This approach is used to construct theories or hypothesis to justify the observed system's behaviours. In this way the simulation model could be used to predict future behaviour of a system based on the changes conducted in the model.

Simulation models can be classified along three separate dimensions static vs. dynamic models, deterministic vs. stochastic models and continuous vs. discrete event model [121]. A static model is a representation of a system at a particular time. In this model time does not play any role. Dynamic models, on the other hand, represent the system as it evolves over time.

Deterministic models do not contain any probabilistic components. The outputs of these models are usually specified once the set of input quantities and relationships in the model are specified. However, stochastic simulation models are models that include some random input components. As a result of random inputs to the model the output of the model will be random as well. The random output of the model must be treated as only an estimate of the true characteristic of the model. Good experimental design techniques are requirement to analyse model outputs.

A continuous simulation model is a system representation over time. The rate of change of the state variables with respect to time are typically based on ordinary or partial differential equations with time as an independent variable. State events, unlike time events, are not scheduled to occur but they occur when a continuous variable reaches a predetermined level. Continuous simulation model can be either deterministic or stochastic, as it is possible to incorporate random events that depend on time, into differential equations.

A discrete model assumes that the state of the system changes only at specific times, which are often refereed to as events and that the state of the system is unchanged between these events. These models are represented by time events. When an event occurs, the simulation model determines the effect of the event on the state variables. These models may be either deterministic or stochastic models. Discrete event simulation is less detailed than continuous simulation but it is much simpler to implement and hence is used in a wide variety of situations.

There are also other ways to distinguish different types of model. Real time simulation is used when the simulation must be interfaced to real hardware or software to a human. This type of models could be used to test hardware or embedded software.

Distributed simulation modelling has gained a lot of interest in recent years. In this approach simulation programs executing on different computing platform interact with each other over a network. This can offer a convenient way to combine several software to form more complex simulations. The emphasis in distributed simulation is often on the reuse ability and interoperability of models.

Application areas for simulation are numerous such as designing and analysing manufacturing systems, reengineering of business process, determining ordering policies for an inventory system and etc. In general simulation is widely used in operation research and management science techniques. [121] reported that simulation was consistently ranked as one of the three most important operation research techniques. Also, [122] in his analysis of 1249 papers found that simulation was the second mostly used approach among 13 techniques considered.

In a simulation study, human decision making is required at all stages, namely model development, experiment design, output analysis, conclusion formulation, and making decisions to alter the system under study. The only stage, where human intervention is not required is the running of the simulations, which most simulation software packaged performs efficiently. The absence of powerful simulation software can hurt a simulation study but its presence will not ensure success. However, Experienced problem formulators and simulation modellers and analysis are indispensable for a successful simulation study.

5.3 Problem suitable for simulation modelling and analysis

In general, whenever there is a need to model and analyse randomness in a system, simulation modelling is the tool of choice. More specifically, situations in which simulation modelling and analysis is used include the following [123]:

- It is impossible or extremely expensive to observe certain processes in the real world.

- Problems in which mathematical model can be formulated but analytic solutions are either impossible or too complicated.
- It is impossible or extremely expensive to validate the mathematical model describing the system due to insufficient data.

In general, simulation modelling is mostly the chosen methodology in many cases due to the sheer complexity of the systems of interest and of the models to represent them in a valid way. As mentioned earlier, the application of simulation abounds in many areas, however the most recent growth in simulation applications has been in the manufacturing area. Almost all major manufacturing process designs currently benefit from some sort of simulation analysis of the process designs. Appendix D provides some points on the advantages and disadvantages of this approach.

5.4 Simulation application in logistics management

The logistics and transportation systems utilise many resources usually classified as direct and indirect resources. The direct resources are used in the physical transportation of freight or goods from one geographical location to another, and indirect resources involved in sorting and consolidating at the various transit locations. The deployments of these resources are to ensure the least amount of delays at terminal, maximum availability and utilisation of resources and on time pickup and delivery of goods. These are some of the challenges that managements are facing in managing the logistics network in an efficient manner for a smooth and balanced operation. A critical role of logistics management is how to make an optimisation decision under an uncertain and noisy information environment. Based on the studies conducted in [124] the logistics management as one of the critical problems that need to be addressed for all efforts in manufacturing. Also [70] argues the complexity of logistics management in the following three main categories:

- **Structure properties:** this is related to the infrastructure in the context of logistics, and covers physical as well as information and communicational structures.
- **The dynamic property:** this is related to the processes performed on the network, i.e. the flow of goods, money and information within the structure and hence the dynamics in these processes.

- **The property of adaptation:** this is related to the organisation and the decision making, the management of the structure and the dynamics, in order to realise the process of satisfying customer demands in an effective way.

Therefore to assist the management in wiser decision-making, both simulation and optimisation heuristic models are needed to meet the challenges of the transportation and logistics/supply chain problems. [69] divide the logistics problems that are appropriate for simulation studies into three major categories:

- New design,
- Evaluation of Alternative Designs,
- Refinement and redesign of Existing Operations,

The optimisation and heuristic approaches are mainly used in addressing problem areas falling under the new design category. However, simulation could be used to verify and validate the optimised new design. Also, manufacturing capabilities, production processes, layout arrangements, resource configurations, alternative strategies and operations run rules can be more dynamically reconfigured according to simulation analysis.

Also, [125] stress the importance of the use of simulation as a critical decision support in the development and improvement of management and logistics within SME sector. They further purpose a new framework for creating policies for lean simulation. The principal benefit of this approach is lead-time reduction, faster attainment of simulation results, reduced cost in simulation development.

Also, [69] proposed a Complex Adaptive System (CAS) as a result of the need for manager to be able to adapt to ever-changing demands of their environment and customer demands. The CAS approach is applied by the use of agent based method and simulation modelling. This approach allows global as well as local behaviours to be analysed and evaluated. Therefore, validation and verification any system being modelled can be made for each agent on micro level.

Furthermore, [126] purposed an integrated framework allowing the logistics and manufacturing activities to be integrated to provide a facility for effective design, analysis and optimisation of integrated logistics chains.

Today, many commercial software packages are being employed by the logistics industries depending upon the level of complexity and size of the problem investigated. These software packages range from standard Linear programming packages such as LINDO, CPLEX, OSL to special purpose software such as INSIGHT, SUPERSPIN, and CAPS tool kit. With respect to commercial simulation software, a large number of vendors provide packages that focus on modelling and analysis of simple material handling systems to complex flow-through centres and transportation networks.

5.5 Simulation application in Supply chain management

Supply chain problems are often very large and complex. Some of the contributing factors to this complexity are such as the interactions between the entities, the length of the supply chain, the lead times of manufacturing and shipping, the complexities of modelling the individual entities and the stochastic nature of the demands. Because of these complexities, very few analytical models exist addressing only simplified versions of the problem, which often are based on limiting assumptions.

The use of simulation as a means for understanding issues of organisational decision-making has gained considerable attention and momentum in recent years. Several researchers [125, 126, 127] agree that simulation provides a much more flexible means to model dynamic and complex networks. Based on [130] "simulation is one of the best means for analysing supply chains because of its capability for handling variability. Also, the ability of simulation to capture the dynamic behaviours of systems makes it a reliable method for studying the performance of the supply chain network

Also, [131] provided an extensive study on the application of simulation within the supply chain management. They specify that the main scope and objective of simulation within the SCM is to support network supply chain design and supply chain strategic decision support.

The design of supply chain network as an integrated network with several tiers of supplies is a difficult task. Simulation can be used as a decision support system within the design phase involving facility location, logistics, production capacity, network and production nodes. There are many approached to tackle this challenge for instance [132] has developed a hybrid simulation optimisation approach to address the supply chain configuration design problem.

Also simulation can be used for node localisation, which involves placing a supply chain node in a determined geographic site. In [133] a simulation model was conceived in order to identify the right geographic disposition for distribution centres, aiming to minimise transport costs through the use of proper cost functions.

In addition, simulation is applied over a supply chain to evaluate more strategic alternatives as strategies based on quick response, collaborative planning and forecasting or outsourcing to third parties. [134] designed and developed an object oriented supply chain simulation framework to facilitate the dynamic analysis of supply chain system. In [135] simulation was applied on a DimerChrysle Corporation's vehicle logistics network from assembly plants to dealers across North America to reduce the order-to-delivery times by increasing network efficiencies across the distribution chain.

Some of the mostly used application of simulation within SCM can be categorised as follows [131]:

- **Demand and sales planning:** simulation processes dealing with stochastic demand generation and forecasting planning definition. [126] used simulation techniques to evaluate the effects of various supply chain strategies on demand amplification. [128] used SCGuru to evaluate the impact of demand and transportation variations on the supply chain.
- **Supply chain planning:** simulation processes supporting raw material sourcing, production planning and distribution resources allocation, under supply and capacity constraints. [127] utilised a simulation to study the effect of sharing suppliers available to promise information. Also [129] developed a simulation based genetic procedure to determine optimal setting for controllable inputs (i.e. sourcing) to the supply chain. To

support supply chain planning [136] proposed an object oriented, scaleable simulation based control architecture. The control refers to automatic triggering of value chain interactions such as request for quotes (RFQs), Purchase Orders (POs), transhipment and resource allocation decisions in the ERP/MRP systems, real-time, based on the conditions perceived in any partner in a value chain.

- **Inventory planning:** simulation process supporting multi-inventory planning: the commercial simulation tool programmed by Promodel, SCGuru, proposes a specific module for inventory management and optimisation [137]. IBM [138] used Asset Management Tool (AMT) to achieve quick responsiveness to its customers with minimal inventory. [139] used Simulation Dynamics' Supply chain builder to study the best strategies for allocating inventory to distribution centres in a nationwide food production and distribution network.
- **Distribution and transportation planning:** simulation of distribution centres, sites localisation and transport planning, in terms of resources, times and costs; it is one of the most recurrent simulation processes reported in literature: for example, IBM Supply Chain Analyser (SCA) has two separated modules (distribution and transportation planning) to simulate distribution centres, transport type and relative management processes (material handling, loading and unloading) [27]. Also [140] developed a stochastic, discrete simulation model of bulk transportation to show the impact of transportation logistics on production performance.
- **Production Planning and scheduling:** Supply chain scheduling plays and important role as indicated in [24]. Poor planning may lead to system instability that could highly impact the ability of the supply chain to satisfy its customer demands. The decision to be taken is defined in most cases based on time frame. Manufacturing planning is implemented by simulation models and tools, which integrate different model layers, from single production lines to entire factory and to the whole logistics chain.

  o <u>**SDI Industry Pro**</u> [142] is one of the most important examples of manufacturing planning implementation; SDI is a simulation tool specifically developed for logistics chains, which allows the development of models from single production machines to more complex distribution centres.

- o **Narayanan et al.** [143] developed an object oriented model consisting of; basic simulation, inventory control, Shop floor, suppliers. This model was developed to assist in enhancing the performance of the supply chain production planning system.
- o Furthermore [24] proposed a supply chain scheduling approach using a distributed parallel simulation. Supply chain scheduling can improve the company's performance when its activities and resources are costly and sensitive to errors and the involved partners require intensive co-ordinations.

Based on literatures simulation technology used in addressing SCM issues can be identified in following categories [131]:

- **Local simulation paradigm:** these are specific commercial simulation tools and general-purpose simulation tools or languages. The commercial tools are those packages developed for simulation purposes within a supply chain context such as SDI Industry Pro, IBM SCA, SCGru in, LOCOMOTIVE, Supply Solver and SIMFLEX ™. Also, Arena, Create!, CPLEX, ModSim and STROBOSCOPE [127] are general-purpose simulation tools or languages.
- **Parallel and distributed simulation paradigm:** Typical examples for such tools are such as: CMB-DIST, MPI-ASP, GRIDS, HLA, and DEVS/CORBA.

5.6 Simulation and Optimisation

The optimisation of simulation models deals with the situation in which the analyst would like to find which of possibly many sets of model specifications (input parameters) lead to optimal performance.

Generally, the input parameters are known as factors and the output performance measures are called responses. In a typical experimental design the aim would be to find out which factors have the greatest effect on a response. However, optimisation seeks the combination of factor levels that minimises or maximises a response subject to the constraints imposed on factors or responses. In the context of simulation optimisation, the simulation model can be considered as a function that evaluates a set of input factors by turning them into output performance measures. According to [129] using simulation in the optimisation process presents a number of challenges:

- In this approach, there is no analytical expression of the objective function, which eliminates differentiations or exact calculation of local gradients.
- The stochastic nature of the simulation results in problems as given a set of deterministic decision variables; the performance measure is not deterministic but rather is represented by a probability distribution.
- Simulation programs are typically computationally more expensive to evaluate than analytical functions.

Also [144] identifies four main classical approaches for optimising simulations such as: stochastic approximation, response surface methodology, random search and sample path optimisation. However [145] suggests while these approaches account for most of the literature in simulation optimisation, they have not been used to develop optimisation for simulation software. [146] justifies this lack of practical commercial implementations as mainly due to the current simulation optimisation methods generally require a considerable amount of technical sophistication on the part of the user, and they often require a substantial amount of computer time.

Alternatively, a number of researchers [147] suggest the use of Metamodels as filters with goal of screening out solutions that are predicted to be inferior to the current best known solution could help to reduce computation time. This is the approach taken here in this work. The GA search engine provides a pool of competitive generated solutions and then the simulation model is used to further evaluate the solutions.

5.7 Simulation Tools:

Manufacturing simulation models can be developed using both general purpose and manufacturing focused tools. However, depending on the complexity of the system being modelled, manufacturing focused tools can significantly simplify and quicken the modelling process. [148] classified simulation languages at three different levels on the basis of the type of system being modelled. Appendix D provides description of this classification.

There are many different manufacturing oriented simulation packages on the market and each has its strengths and weaknesses. In selecting the appropriate simulation software for and application, there a number of metrics that could be used to evaluations such as [123]: Modelling flexibility; Ease of use; Modelling structure (hierarchical vs. flat, object oriented vs. nested); Code reusability; Graphic user interface; animation, dynamic business graphics; hardware and software requirements and finally statistical capabilities, output reports and graphical plots, customer support, and documentations.

Some packages focus on ease of use and compromise flexibility, while others focus on flexibility and are more difficult to use. Because most manufacturing systems have some unique intricacy, the best packages allow the user to combine easy to use constructs with more flexible, lower level constructs. In this application Witness Simulation software was used to develop models from considering supply chains. The following section briefly describes this modelling tool.

5.7.1 WITNESS Simulation Software

Witness is a comprehensive discrete event and continuous process simulator. Witness allows one to rapidly, incrementally and accurately build, debug, validate, verify and exercise complex models. Witness provides a number of features helping to reduce model-building time these are such as: The use of colour coded status icons for each element, It provides partial or complete usage of a built model, It provides a user defined library (designer elements), Graphical interface, Pre-defined input/output rules and Built in error checking facility. In addition, modelling accuracy is supported through the creation of an action language, a host of system and user-defined functions, user interactivity and object linking and embedding (OLEII).

5.7.1.1 Building A WITNESS MODEL:

Figure 5-1, shows the model development interface of the Witness simulation tool. In general, Witness models are built in three steps such as define the elements to be included in the model, display the necessary elements and finally detail the elements.

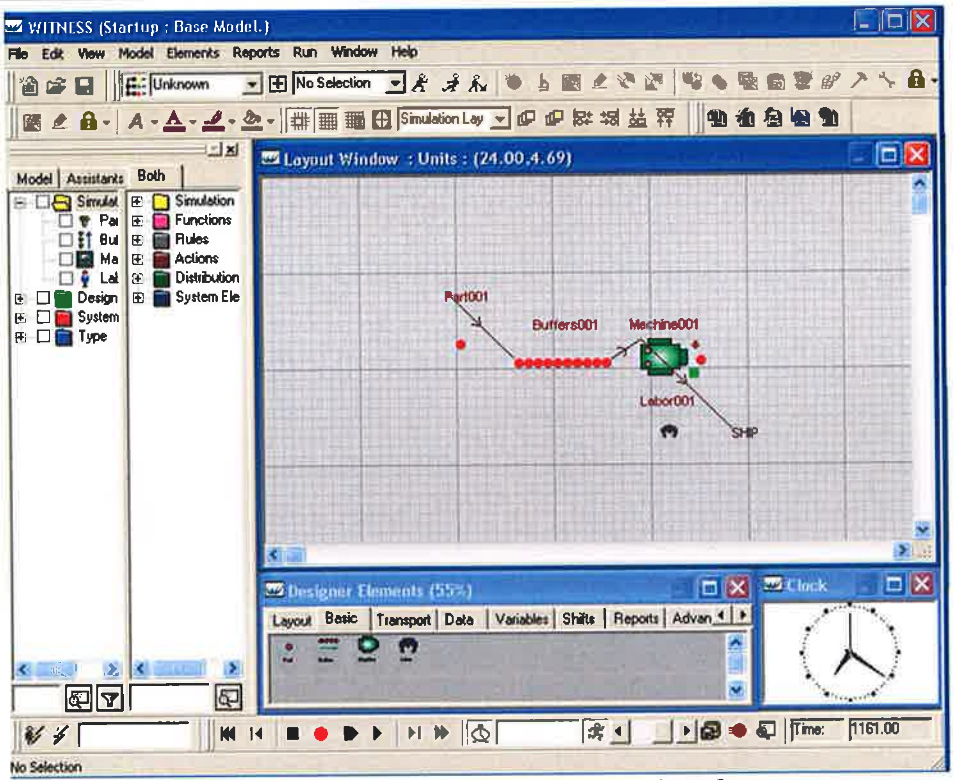


Figure 5-1 Witness Model development interface

- **Define Model Element:** To construct a simulation model, the user must first specify the type of elements to be used within a model. The user could use the define mode through a menu bar or select the element type to be defined, and provide a name and quantity required for the element. Witness elements fall into 3 categories:

- **Physical:** WITNESS provides elements that represent tangible objects. The physical elements are categorised as follows:

    - Elements that move through the model, being processed (parts and fluids).
    - Elements that transport parts and fluids (conveyors, track, vehicles, carriers, paths and pipes).
    - Elements that store parts and fluids (buffers and tanks).

- **Logical:** These elements provide the modeller a greater control over execution of the model. Logical elements include: Elements that represent shift patterns (shifts), Elements that provide statistical variation (distributions), Elements that provide a source of data outside the model (files), Elements that are referenced in the model's control logic (variables, attributes and functions).

- **Reporting:** This allows the user to select information to be graphically displayed while the model is running. The element used for this purpose are such as histograms, pie charts, time series.

- **Displaying Model Element:** Elements are displayed either from the display option from the menu bar or by bringing up the display form for that element as shown in Figure 5-2. Once an element is selected, a sub-menu specific to the element type is presented to place the chosen item accordingly. Also, the user is provided with an option to an individual or group of elements to a new position.

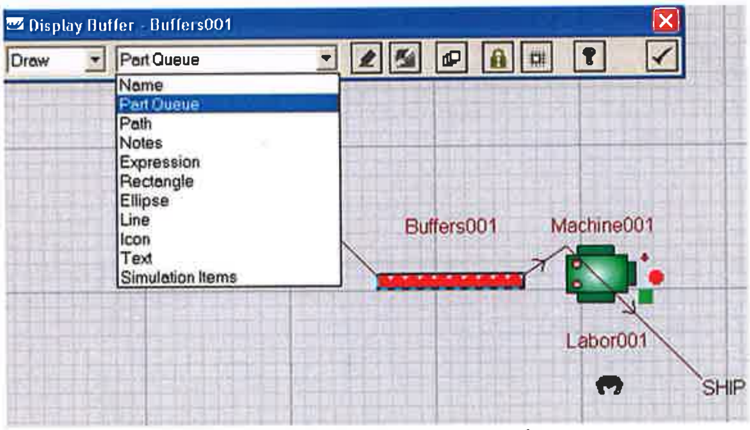


Figure 5-2 Display menu option

- **Detailing Model Elements:** each element type has specific detail form as shown in Figure 5-3. This involves entering details such cycle times, setup conditions, capacities and etc. this option can be accessed either from menu bar or from the element itself. Detailing the element include the use of Witness Input/Output rules and Action Language. Witness provides a set of predefined rules for pushing or pulling parts/entities through the model. At a basic level, elements such as machines/activities would either pull from within the input rule editor or push from within the output rule editor. Also, IF/Elseif logic or user defined functions may be employed to move the parts/entities. Witness also provides with a construct known as the Action Language. This construct allows the modeller more flexibility in describing elements behaviours. Syntaxes such as FOR/NEXR and WHILE/END within this construct allow the user a greater control over model creation.

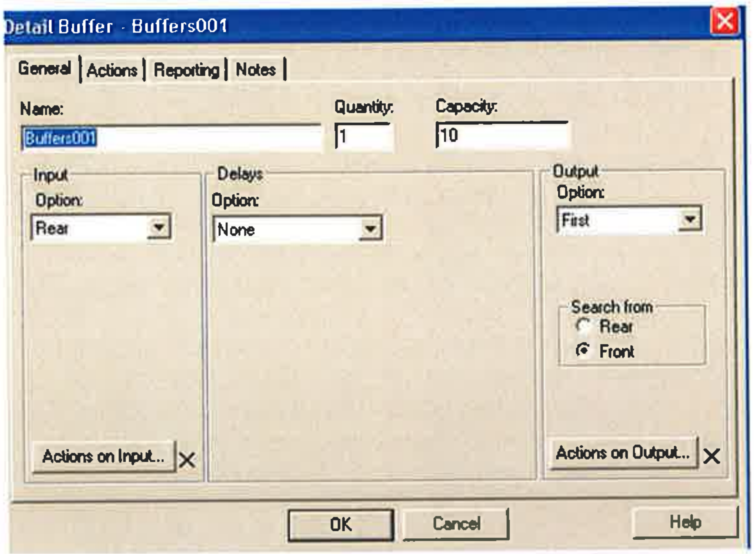


Figure 5-3 Detail Menu option for a buffer element

The reader is asked to refer to Appendix D for further descriptions on Witness simulation tool and its features.

5.8 *Alternative Approaches to simulation Modelling*

The recent advancements in computer technology have provided researchers with enabling levers to address complexities with applying simulation modelling. The following section briefly covers some of these techniques.

5.8.1 Object Oriented Simulation Modelling

Object Oriented Programming (OOP) concept could help us greatly to comprehend and manage the complexity of the system. For a design to be object oriented, it must have the following characteristics:

- ***Abstractions:*** An abstraction denotes the essential characteristics of an object that distinguishes it from all other kinds of object [149]. Abstraction allows focussing on those properties of the object that are relevant to the problem and ignoring the irrelevant properties.
- ***Encapsulation:*** This involves hiding all of the details of an object that do not contribute to its essential characteristics.
- ***Modularity:*** This is grouping logically related abstraction into modules. This is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules.

- ***Hierarchy:*** this is ranking and reordering of abstractions. Hierarchy is about organising related objected in terms of their hierarchical relationships to each other.

A number of researchers [149] suggest that the use of OOP design and modelling would greatly assist in enhancing flexibility, portability, understand ability and maintainability in simulation modelling.

5.8.2 The Parallel, Distributed and Web based Simulation Modelling

Distributed Discrete Event Simulation (DDES) refers to the execution of a single discrete event simulation program on a parallel computer. In this method the considering system is viewed as being composed of a number of Physical Processes (PS) that interact at various points during the simulation run. The interaction between PS is performed by exchanging event messages. As there may be relations between events, therefore sequence of event occurrence is of paramount importance. There are two major classical approaches to control the sequencing of event occurrences known as conservative and optimistic approaches.

The conservative approach prohibits any concurrent execution of events, where as optimistic method uses detection and recovery approaches. In the latter method, the errors are detected and a Rollback mechanism is used to recover. The Time-Wrap mechanism is the most well known optimistic approach.

The modern logistics introduces a very complex reality that requires a high level of coordination among many different entities such as sites, operators and etc. [150] proposed and developed distributed simulation approach to model the complexity of the modern logistics systems.

The developments in the Internet and World Wide Web (WWW) have also offered a significant potential in overcoming the hardware and software problems with the interactive simulation methodology. [151] described a Java-Based Architecture for developing Interactive Simulations on the Web.

5.8.3 Agent Based Simulation Modelling

Agent based modelling and simulation (ABMS) is relatively a new approach that has found application in many different fields. In this approach a simulation experiment is constructed around a set of agents that interact with each other and their underlying environments to capture the real-world behaviours. The agents represent the following characteristics [152]:

- ***Autonomy:*** Agents operates without any direct intervention to control their actions and internal states,
- ***Reactivity:*** Agents perceive their environments and respond to changes respectively,
- ***Pro-activity:*** Agents may not only react but can depict a behaviour to fulfil their own goals,
- ***Social Ability:*** Agents interact with other agents and/or human beings either using explicit or implicit communication.

In addition, [153] reported the use of an ABMS technology to model, design and simulation global distributed supply chains. They suggested ABMS provided a short model development time with reduced human resources. Also, they agree that ABMS provides a powerful approach to life cycle support of the supply chain information architectures. Also, [108] described the development of ABMS for assisting in decision-making regarding supply chain management. In addition [155] used ABMS in logistics management

The ABMS characteristics provide researchers with a better ability to explore dynamic behaviours of a system, which cannot be obtained, based on any analytical or mathematical methods at the system level.

5.9 Conclusions

Simulation can be used as an evaluation tool, providing for the collection and evaluation of quantitative system performance measures. It can also be used as an analysis tool in experiments examining alternative designs and operating strategies on system performance. However simulation as an experimental tool does not solve a problem or optimise a design. It supports in evaluating a solution and provides understanding of problematic areas rather than generating a solution. An optimum solution can only be obtained through experimentation by running and comparing the results of alternative solutions.

The logistics management is perceived as a complex task due to it structure such as physical, information and communication. Also due to its dynamic property related to the flow of goods, money and information within the system and finally due to continues customisation to meet customer demands. Also some of the contributing factors to the complexity of supply chain management are due to factors such as the interactions between the entities, the length of the supply chain, the lead times of manufacturing and shipping, the stochastic nature of the demands.

To assist managers in wiser decision-making, both simulation and optimisation heuristics are needed to meet the challenges of the transportation and logistics/supply chain problems. Simulation modelling provides a more flexible means to model dynamic and complex networks. It captures systems' variability and dynamic behaviours of the systems more efficiently. Integration of this tool with optimisation tool could lead to search for optimal performance.

In the context of simulation optimisation, the simulation model can be considered as a function that evaluates a set of input factors by turning them into output performance measures. The potential benefit in using optimisation simulation integration is the reduction of computation time and technical sophistications. The proposed DSS architecture supports this principle. To this effect the following chapter aims to describe how a simulation model was developed from the considering supply chain. This is the second module of the proposed DSS.

Chapter 6: Simulation Model Development and Validation

6.1 Introduction to Managing the Supply Chain

In manufacturing, the supply chain is the linkage for the physical movement of all materials from suppliers, through transformation, and then as finished goods for the customer. In service, the supply chain is distribution, where the start point is the finished product that has to be delivered to the client in a timely manner. The purpose of this chapter is first to describe different components of the considering SME in terms of common supply chain principles. This is followed, by illustrating how a simulation model was developed for this system. For this reason different component of this model are described and it is shown how different logics and specific details are modelled. Furthermore, the model verification and validation process are presented. The principal aim in developing this simulation model is to provide an evaluation tool to further analyse and evaluate the generated near optimum transportation schedule from the genetic search engine module. This module is aimed to provide higher credibility for transportation schedules and therefore assisting the decision makers.

6.2 Activity in supply-chain management

Management of supply chain involves rigorous attention to quality, cost and lead or delivery times. It implies teamwork, cooperation and effective coordination throughout the entire organisation. Some key management activities include: Site selection, forecasting, and development of an operations plan, raw material management and purchasing, distribution requirements. Appendix E, describes the considering supply chain based on these activities. The reader may refer to this appendix for familiarity with supply chain concepts used to describe this system.

6.3 Steps of a simulation project

When conducting a simulation study, it is recommend that a structured systematic approach to be carefully planned and rigidly adhere to. The "40-20-40" rule is a widely quoted rule in simulation studies [156]. The rule states that in developing a model, an analyst should divide the time as follows:

- 40% to problem definition, project planning, system definition, conceptual model formulation, preliminary experiment design and input data preparation.
- 20% to model translation, and
- 40% to model validation and verification, final experimental design, experimentation, analysis and interpretation, and implementation and documentation.

6.3.1 Data collection and Preparation

This study is mainly concentrating on addressing vehicle routing and scheduling to reduce the transportation cost. This could greatly impact the factory performance as discussed in Appendix E. Once the fleet routing and scheduling had been identified, as the area on which the study should concentrate, and the preliminary objectives for the study had been determined, the next step was to gain familiarity with system. This was conducted in different ways as follows:

- Discussing the system with Logistics managers, Bottling plat operator, supervisors and managers. Also, different ideas were exchanged with truck derivers as the main resources in LPG transportation.
- Identifying the key sections in the supply chain and gaining knowledge about how they operate and interact with each other.
- Identifying the LPG Inventory management methods, customer service level and selection of distribution channels.
- Discussing delivery scheduling and vehicle routing.
- Identifying personnel who have knowledge of, or access to key information.
- Documentation of all information gained for further reference.

After an acceptable degree of familiarity with the system had been achieved, the next step was to collect and document all the data need for establishing a successful construction of a computer simulation model of the system. Information were mainly obtained from such sources as personnel, the quality documentation, vehicle routing and scheduling sheets, industrial engineering records, daily LPG production reports and machine maintenance records. This process was proved to be a time consuming, requiring six months to complete fully.

As the simulation model was developed to evaluate the impact of vehicle routing and scheduling on the LPG supply chain, the main issues tackled in developing this model was as follows:

First to determine systems operational constraints at different levels such as supply ports, transportation fleets, processing plants and clients. The existing operation constraints in the real system highly influence the manager's decisions on performing routing and scheduling for distribution activities within the supply chain.

The second issue to be tackled was to identify those processes within each level that could be influenced or could affect the vehicle routing and scheduling. To this effect the main operation rules were studied at each level. This process involved analysing information from the industrial engineering department, to identify how to model cycle times for each operation (i.e. filling, queuing, inspection, delivery rate, demand patterns and etc.) within the considering level. This process firstly, reviled that the complexity in modelling this supply chain is mainly due to variability of availability concerning, manpower, vehicles, environment, and other factors. Secondly due to the low degree of automation at supply ports, transportation and bottling plant, there exists a high variability in the system, which results in a great degree of uncertainty in vehicle routing and scheduling decision-making.

In some cases involving automated machine operations, the standard times were used to model the machine operations. In other cases representing high variability's such as transportation times and customer arrivals data were collected and information on production histories, breakdown and repair histories, set-up histories, delivery and demand patterns were analysed.

6.3.2 Model Development

After, the initial data collection and preparation, the model development process was started. This attempt was soon discarded, as it was found that not enough was known about the system in order to develop a simulation model of it. This attempt helped to identify more precisely the type of data and knowledge required for the model development and the formats that the data should be in.

When the data collection was completed, it was decided that enough knowledge had been accumulated about the system and therefore, the model development was restarted. The simulation model developed based on operations performed in LPG supply ports, Transportation fleets, Bottling plants and client demands. These levels are described in the following subsections.

6.4 Simulation Modelling Of The Supply Nodes

The LPG is supplied from refineries dispersed in different geographical localities. Each refinery in the system has a specific supply capacity that could be offered to the companies. The refineries work on 2 shifts to supply the customers with LPG all the time. However, there are always uncertainties in supplying the required amounts due to many factors such as breakdowns, overhauls, and industrial actions and so on so forts. As shown in Figure 6-1, to model the refineries a number of objects were used, which are described as follows:

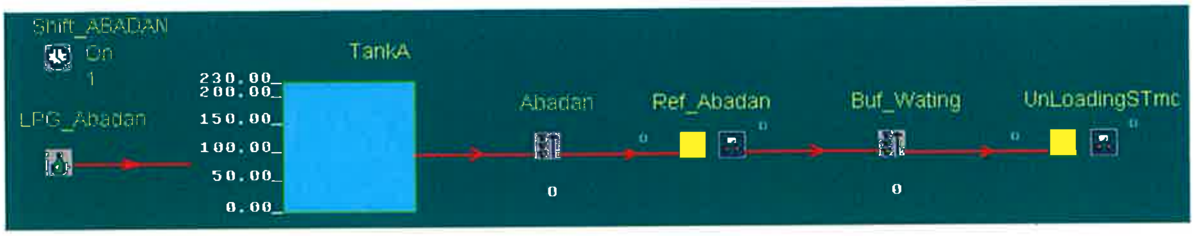


Figure 6-1 Modelling Refineries in Witness simulation environment

- **LPG:** To model LPG in the simulation model, the fluid object was used from the object library. A fluid is a continuous element that could be used to represent liquids that flow between processors, pipes, and tanks. Fluids could change type or colour as they pass through processors, tanks and pipes. In this simulation model there are five main types of fluids used to represent LPG produced by different refineries. Every type is used within a refinery. Each type has its own fluid colour. This is done to distinguish the source of supply of the LPG in the simulation model. There is also, a dummy fluid type called LPG to set the initial fluid type for the bottling plants' reservoirs.

Figure 6-2 the main types of LPG fluid in the model

- **<u>LPG Reservoirs:</u>** The reservoirs are mainly represented by Tank object from the library. The main parameter of this object to be adjusted is the tank's capacity, which suggests the total volume of LPG that could be supplied to the customers. The following table suggests a monthly supply of LPG available from each refinery in the system. As illustrated in Table 6-1, the suggested volumes are subject to changes due to seasonal factors.

Table 6-1, the LPG supply pattern from refineries in the year 2003 (Tones)

Year 2003	1	2	3	4	5	6	7	8	9	10	11	12
Refineries	January	February	March	April	May	Jun	July	Agust	September	October	November	December
Tehran	600	600	600	600	350	350	200	200	200	350	600	600
Mashhad	430	495	683	568	651	1871	1964	1938	900	651	686	250
Golestan	900	900	500	996	944	881	881	881	881	907	908	900
Esfahan	3644	3396	3884	2921	2759	2434	2305	2325	2758	2576	2492	3720
Arak	600	600	800	600	600	600	600	600	604	590	600	600
Abadan	367	367	367	367	367	367	367	367	367	367	367	367
Bandar Abas	1973	1842	1666	2074	2191	859	1047	1053	1654	2159	2097	1977
Sums	**8514**	**8200**	**8500**	**8126**	**7862**	**7362**	**7364**	**7364**	**7364**	**7600**	**7750**	**8414**

To find the best fit to describe monthly supply provided from each refinery, a study was conducted to collect historical data for the last 10 years. These data are presented in Appendix F. The best distributions to fit the available data are obtained to be used in the simulation model.

Using the Tank object, one must specify the initial fluid type and initial volume of the fluid to be used within this object as illustrated in Figure 6-3. It is assumed that the capacity of the tank and its initial volume are set to the monthly-allocated LPG supply for each refinery. As mentioned earlier, the statistical distributions are used in the simulation model to specify the LPG capacity provided on monthly basis by the refineries in the system. These volumes are to be used within a month period.

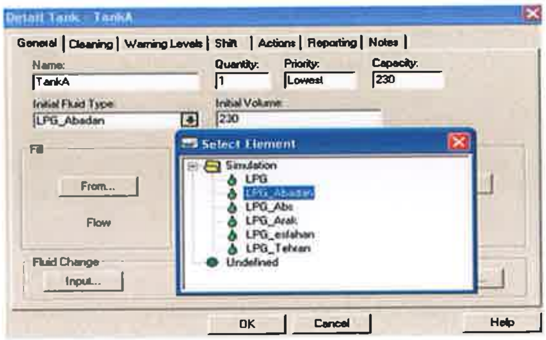


Figure 6-3 Tank object detail property page

- **<u>Refinery's Trucks inlet:</u>** Upon the arrival of trucks into the refiners, they are queued in a waiting area for loading LPG. This is simply presented by a buffer object in the simulation model. Based on the available historical data provided the minimum and maximum waiting times were obtained for each refinery. Table 6-2, represents these waiting times in minutes.

The buffer behaves as a delay buffer for trucks that have been waiting for the minimum time, and as a dwell buffer for trucks that have been in the buffer longer than the maximum time. The minimum and maximum time are both calculated from the time trucks enter a buffer. The capacity of this waiting area does not leave any constraints on the overall LPG loading operations. Therefore the capacity of this buffer is set to a large possible number.

Table 6-2 Average monthly minimum and maximum waiting times in minutes for refineries

Refineries	January		February		March		April		May		Jun	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Abadan	300	1440	240	1080	210	900	240	720	231	924	184	1298
Arak	324	1061	339	1065	322	1373	203	1067	296	1142	288	1317
Bandar Abas	327	1332	311	1290	291	1276	140	300	185	901	210	1205
Esfahan	300	1440	266	1026	272	1286	180	1200	272	1118	197	1047
Tehran	334	1013	328	1022	243	1011	206	994	250	924	222	1291

Refineries	July		Agust		September		October		November		December	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Abadan	180	1200	133	925	148	950	300	1440	163	971	151	1132
Arak	165	731	166	910	153	1000	211	787	144	781	121	982
Bandar Abas	125	1012	142	799	150	811	212	568	121	680	206	1035
Esfahan	150	868	172	1045	138	540	234	1006	124	1178	159	697
Tehran	141	708	161	826	130	648	141	885	191	957	150	658

- **LPG Loading Operation:** Loading the supper trucks are carried out using a single machine object. The trucks are loaded one at a time. The loading time is established based on the historical data provided. The triangular distribution was used to model this time as indicated in Table 6-3. The loading operation would take place during the shift hours; otherwise, the truck must wait in the buffer before, being processed.

Table 6-3 loading time in minutes

Refineries	Average Loading Time (Minutes)		
	Min	Max	Mod
Abadan	120	300	210
Arak	60	300	180
Bandar Abas	120	300	210
Esfahan	60	300	120
Tehran	120	300	210

Usually, there are uncertainties concerning the refineries operations. The main sources of uncertainty are coming from refineries' scheduled overhauls or unpredicted breakdowns. The scheduled overhauls are mainly preplanned and the logistics managers are informed in advance. In modeling the overhauls the shift object could be used. The working hours of refineries are modeled using a shift object in the simulation model. In the case of overhauls the user could set the working hours to the rest hours to make sure that the refinery is not operational.

There could be many reasons for refineries' breakdowns such as mechanical failures, industrial actions, lack of resources and etc. Using the historical data available on refineries' operational failures two parameters were established, time between failures and time to repair. These are provided in Appendix F. Based on the collected statistics, the best distributions to fit these parameters were found as indicated in Table 6-4.

Table 6-4 Statistical distributions used in the simulation model

Refineries	Mean Time to Failure MTTF (min)	Mean Time To Repair MTTR (min)
Abadan	Lognormal(22047.3,15008.5)	Gamma(1.95,89.61)
Arak	Lognormal(21090,15642.8)	Weibull(2.36,250.55)
Bandar Abas	Gamma(2.25,12259.3)	Weibull(1.63,133.44)
Esfahan	Lognormal(21376.4,15643.2)	Triangular(60, 186, 450)
Tehran	Gamma(3.84,5705.5)	Triangular(21,126,318)

- **Trucks Dispatching Area:** After loading operation is performed, trucks are queued in a waiting area ready for departing to a bottling plant for unloading the LPG. Trucks are dispatched to its next destinations based on its transportation schedule.
- **Refinery Shift Hours:** Each refinery in the system has its own shift pattern. These working patterns are modelled using a shift object for each refinery. The loading operation is only performed when the shift object is active; otherwise, no truck is processed.

6.5 Simulation Modeling of the Bottling Plants

At the moment, there are eight bottling plants in this LPG supply chain company. The plants are responsible to provide the daily demands of LPG in their respective local markets. The plants are equipped with gas storage facilities and gas filling stations to meet the daily demands. The plants have different personnel needs, capacity levels of equipments and transportation needs. This is highly related to the potential of their respective local markets for the LPG. To model these plants, different processes performed from delivery of raw material to finished products are studied. Figure 6-4, illustrates the overall operations in a typical bottling plant. These processes could be described in three sections as follows:

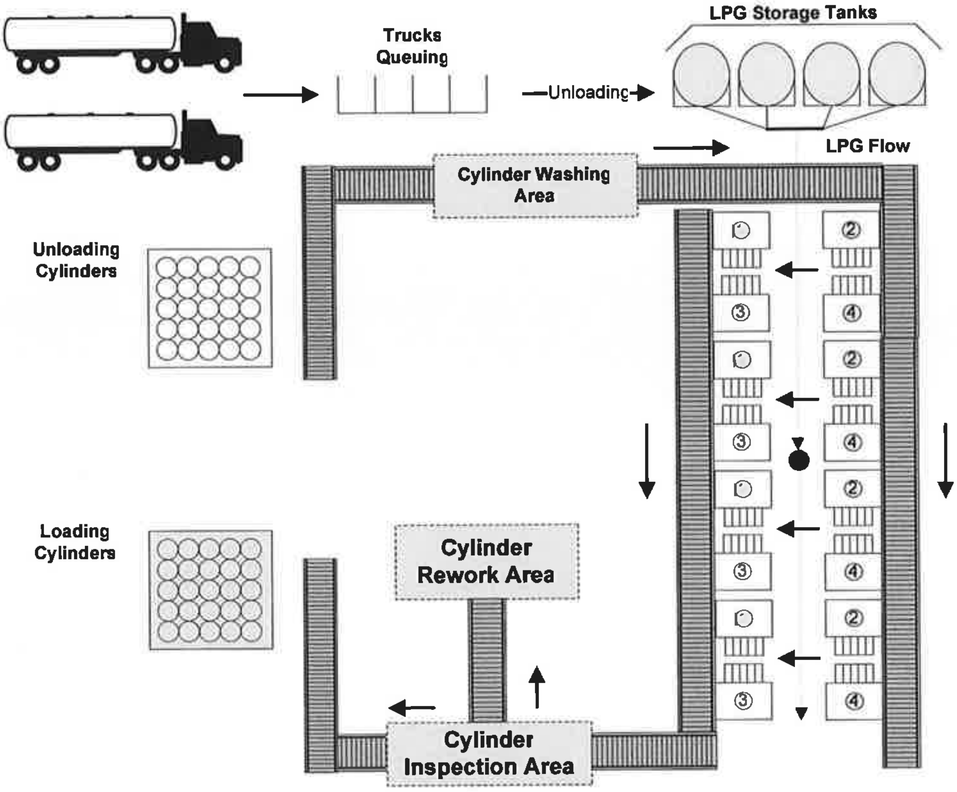


Figure 6-4 Schematic View of processes in a Bottling Plant

6.5.1 Raw Material Inlet

As shown in Figure 6-5, once a delivery of LPG is made to a bottling plant the following operations take place:

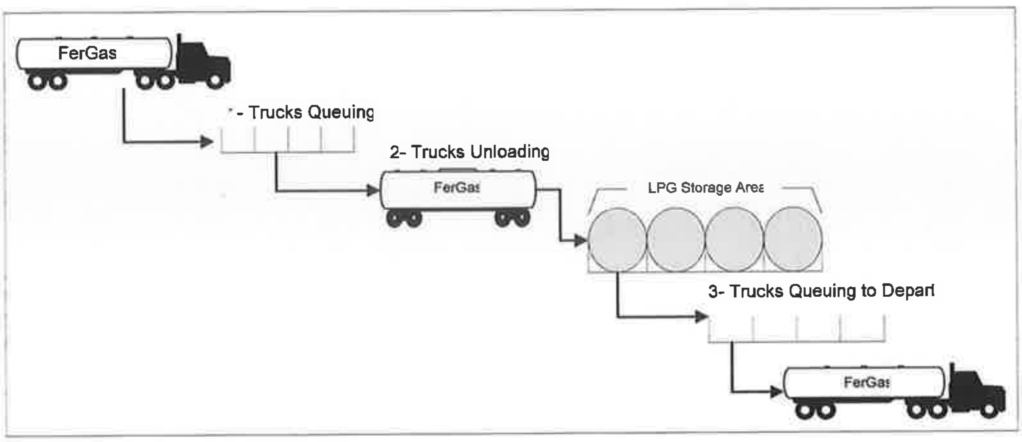


Figure 6-5 Schematic view on Raw Material inlet processes

The following objects from witness simulation tool were used to model the above operations:

- **Trucks Queuing Area:** This is considered to be the first place in a bottling plant where loaded trucks have to wait before being processed for unloading. This area is restricted in terms of the total number of trucks that could be waiting there. Usually, the capacity of this area would allow up to four trucks to stay for unloading. The waiting time depends on the unloading rate of trucks. This area is always open to receive trucks for unloading. Table 6-5 illustrates the queuing capacities for different bottling plants in the system.

This area is simply modelled, using a buffer object from the Witness object library. Here, the buffer represents a space containing trucks waiting to unload. The capacity is a parameter used to specify the total number of parts that the buffer could hold. This parameter is set based on any space restrictions that exist in the respective bottling plant.

Usually, trucks arriving at bottling plats are served as first come first served bases. Therefore, the buffer Output rule's option is set to First truck, searching from the front of the buffer. Also, in modelling this queuing area, it is assumed that there are no time limitations on accepting trucks for unloading. Therefore, trucks arriving at any time during the day are processed based on their order of arrivals.

Table 6-5 Queuing capacities for bottling plants

No.	Bottling Plant	Queuing Capacity
1	Babol	4
2	Esfahan	4
3	Karaj	6
4	Kashan	2
5	Kerman	3
6	Mashhad	4
7	Qom	2
8	Yazd	2

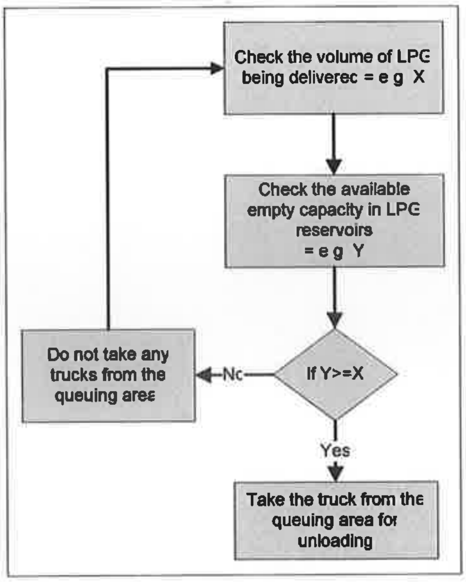


Figure 6-6 Unloading decsion process

- **Unloading Operation:** Trucks are usually unloaded trucks one at a time. There is no time restriction to perform this operation. Therefore, trucks are unloaded as soon as they arrive in the plant. There is always a need for skilled personnel to mind all the safety aspects when unloading is being performed. As a result there is always an operator available to perform this operation. Usually this operation would take some time between one to three hours. It is important to notice that this operation would take place only if there is enough empty capacity available in the storage area of the bottling plants. Otherwise, the truck must wait in the queuing area. The unloading operation is modelled using a machine object.

Witness provides seven types of machine to model different types of processing such as Single, Batch, Assembly, Production, General, Multi-cycle and Multi-station machines. A single machine object was used to model trucks' unloading operation. This is a discrete element that takes in trucks; unload LPG from these trucks and sends trucks to their destinations.

Trucks are unloaded only one at a time, when the availability of free capacity within reservoirs is checked. Any time that there is a change in the state of LPG reservoirs a check is made to make sure that there is enough available capacity to let a truck in for unloading operation as shown in Figure 6-6.
Using, the Fluid emptying rules the LPG is uniformly unloaded amongst the existing reservoir tanks. It is also, assumed that this operation could take place at any time during a day. As there was no restriction on the availability of skilled personnel to perform this operation, therefore, resources/ labour assignment to this operation was ignored.

To measure the unloading time, data were collected from different bolting plants and a distribution was formed to fit these data. The unloading time was modelled using the triangular distribution with the main parameters as shown in Table 6-6. Appendix F provides data used for this purpose.

Table 6-6 Unloading times at bottling plants

	Unloading times (minutes)							
	Babol	Esfahan	Karaj	Kashan	Kerman	Mashahad	Qum	Yazd
Min	76	68	91	86	81	91	86	87
Averag	122	115	118	135	107	592	119	121
Max	174	175	143	180	134	1120	148	157

- **LPG Storage area:** Every bottling plant is equipped with storage facilities for LPG. The capacity of these storage tanks depends on the gas consumption rate of the bottling plant. Therefore, bottling plants have different storage capacities, which highly depend on the local demands for LPG. The volume of LPG to be kept in storage areas is highly seasonal. The maximum allowed capacity of storage area to be used in summer time is about 85% of the total storage area and in the wintertime this is about 95%.

Table 6-7 LPG reservoir capacities provided by bottling plants

No.	Bottling Plant	Capacity (tonnes)	Summer 85% (tonnes)	Winter 95% (tonnes)
1	Babol	120	102	114
2	Esfahan	132	112.2	125.4
3	Karaj	360	306	342
4	Kashan	50	42.5	47.5
5	Kerman	120	102	114
6	Mashhad	120	102	114
7	Qom	50	42.5	47.5
8	Yazd	60	51	57

Witness's Tank object was used to model the LPG storage areas. The trucks contents are unloaded to these objects. Tanks are passive elements in that they do not carry out any process operations on the fluids contained within them. This object allows one to specify warning level for both raising and falling level of LPG in the reserves. Any violations of these levels are recorded for further analysis and evaluations.

- **Truck Departure Area:** This is where the unloaded trucks are waiting for departure to their next destinations. This area is represented using a buffer object. The trucks departure is an operation, which is modelled using a single machine object. This object checks the destination attribute of each truck queuing in the waiting area. This attribute is based on the externally generated schedule for the truck. The trucks are pulled from the waiting area and they are dispatched to their next destination based on their respective transportation schedule.

The following figure illustrates the overall view of the simulation model showing the above processes described in modelling the raw material inlet for a bottling plant.

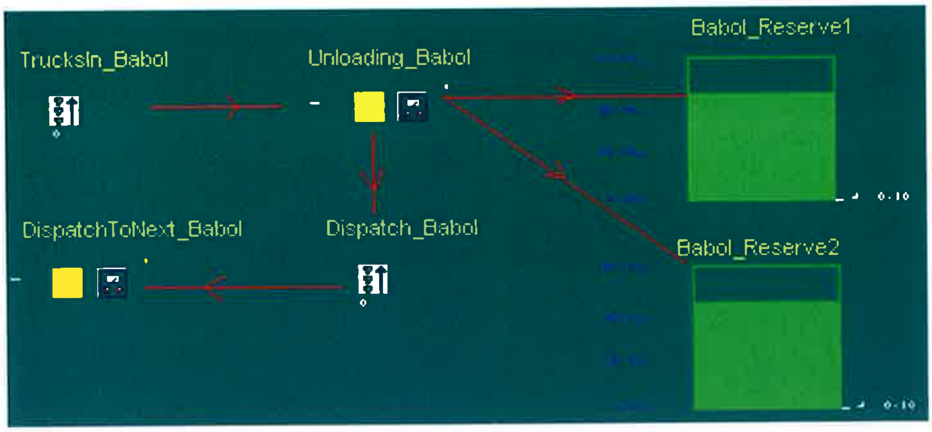


Figure 6-7 Simulation Model view of raw Material Inlet operations

6.5.2 Customer Arrivals:

The company's local representatives are considered to the end customers in this study. The local representatives are responsible to perform local LPG deliveries. As mentioned before the customer's demands are highly seasonal. Figure 6-8 shows the annual customer demand variations for LPG cylinders.

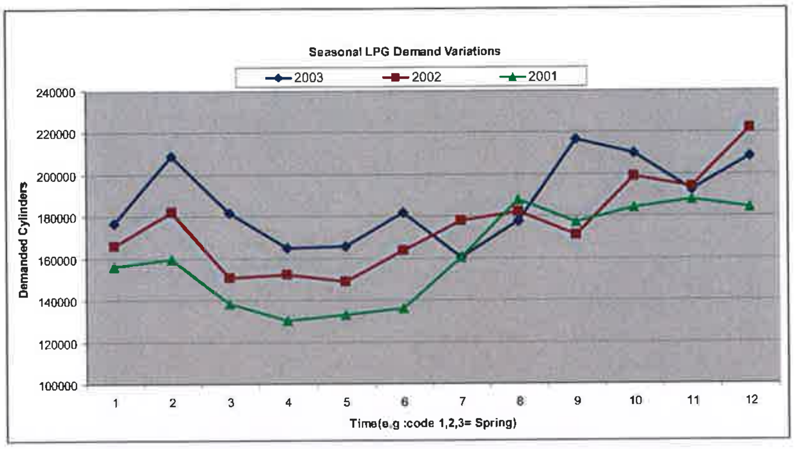


Figure 6-8 Seasonal customer demands [code 1, 2, 3 = spring, code, 4, 5, 6 = summer and etc.]

The representatives would arrive to bottling plants during the day to deliver empty cylinders and to pick up filled cylinders for delivery to the end customers. Usually there is a higher flux of customers in the mornings and the rate of customer arrival decreases along the day as demonstrated in Figure 6-9.

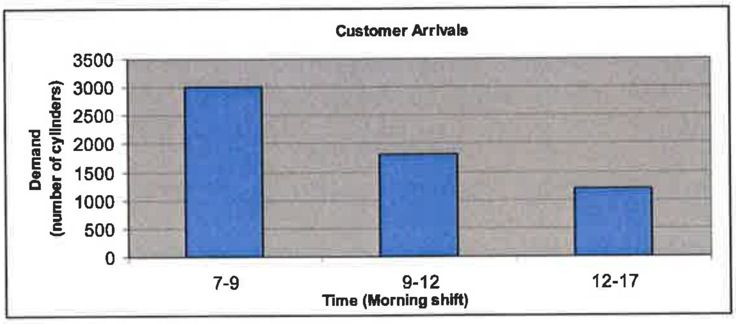


Figure 6-9 Arrivals of customers to a bottling plant

- **Unloading Cylinders:** Upon arrivals of customers, the returned cylinders are unloaded to a delivery zone in the bottling plant. The unloading time is variable and it usually depends on the number of manpower that is available for this operation.
- **Loading Cylinders:** Once the returned cylinders are unloaded, the customer demands are provided from the already filled cylinders that are stocked in the dispatching inventory. It is important to mention that the policy of this company is based on the make to stock principles due to the high uncertainty that exists in the system. Therefore, there is always an inventory of finished goods to be taken when it is needed. Table 6-8, illustrates the stock capacity of the finished cylinders available in each bottling plant. Furthermore, the loading time is variable and it depends on the number of operators that are available for this operation.

Table 6-8 Dispatching Stock Capacity Available in each bottling plant

No.	Bottling Plant	Dispatching Stock
1	Babol	2000
2	Esfahan	5000
3	Karaj	2500
4	Kashan	1250
5	Kerman	1380
6	Mashhad	1250
7	Qom	1250
8	Yazd	1200

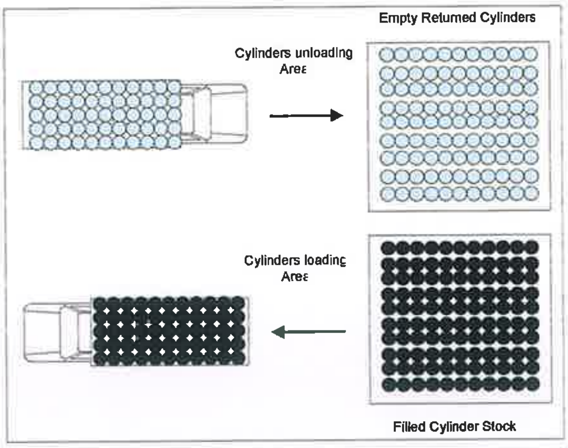


Figure 6-10 Customers Arrivals to Bottling Plants

To model customer arrivals to bottling plants, one must consider the daily working hours for each bottling plant in the supply chain. Normally, the bottling plants operate on a single daily shift pattern. However, based on the seasonal demand for LPG and its raise during the winter time the production manager could further extend the working hours to encounter the second daily shift.

In Witness, a shift is a logical object that could be used to create a shift pattern or a series of shift patterns, which are in effect a sequence of working and non-working periods. As shown in Table 6-9, there are three different shift patterns that are used in modelling the working hours of bottling plants in the system.

Table 6-9, Daily working shift patterns incorporated by different bottling plants

Daily Shift Pattern 1			Daily Shift Pattern 2			Friday Shift Patterns		
Periods	Mode	Duration (Minutes)	periods	Mode	Duration (Minutes)	periods	Mode	Duration (Minutes)
1	Working	390	1	Rest	390	1	Rest	480
2	Working	270	2	Working	270	2	Working	360
3	Rest	30	3	Rest	30	3	Rest	600
4	Working	180	4	Working	240			
5	Rest	30	5	Working	90			
6	Working	240	6	Working	120			
7	Rest	30	7	Rest	300			
8	Working	210						
9	Rest	60						

Using the shift patterns the customers are generated on daily basis. To find the daily demand/ customers for a bottling plant a distribution is developed. This distribution is based on the historical data available from the daily demand for LPG in a specific month. Appendix F illustrates an example of these data.

At the start of the shift, the total daily demand for that day is generated. This value represents the total number of LPG cylinders that are brought to a bottling plant for fillings. These gas cylinders are modelled as follows:

- **Part Object:** In this model, the gas cylinders are presented as Part objects. In Witness, a part is a discrete element that could represent anything that moves between other discrete elements. There are different types of parts such as: Passive, Active and Active with profile parts. The gas cylinders are modelled using the active with profile part object. Using this object, the gas cylinders can arrive into the model in an irregular pattern that could be repeated over a 24 hours period. Using the detail dialog of this object, one could set up a part arrival profile that could contain a repeating pattern as illustrated in Figure 6-11. The arrival pattern of customers is modelled based on the historical data available on this for each bottling plant. Appendix F presents these data.

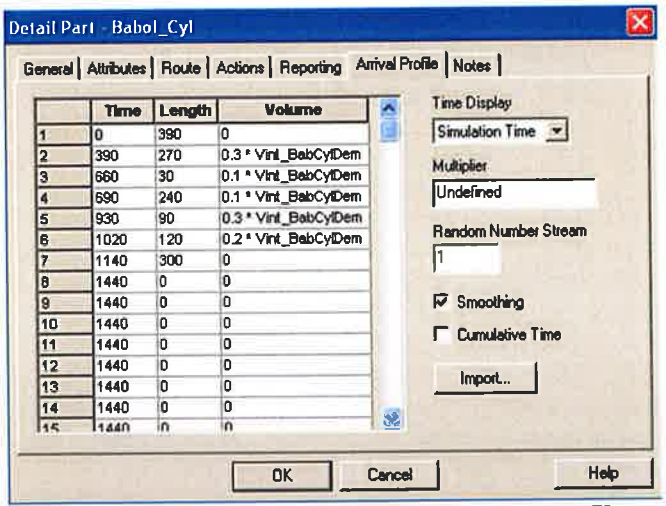


Figure 6-11 Gas Cylinder Part with arrival profile

6.5.3 LPG Cylinder Processing:

The returned cylinders are manually placed on a conveyor to be taken for different operations in the plant. Figure 6-12 illustrates operations involved in processing the cylinders. The following describes briefly how they are modelled in this application.

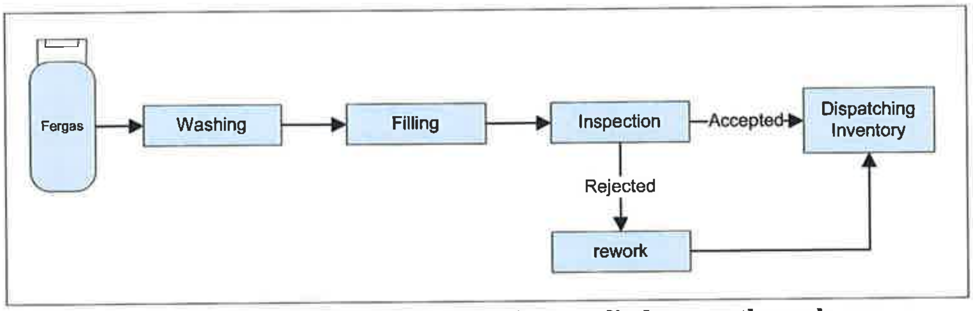


Figure 6-12 Different processes that a cylinder goes through

- **<u>Cylinders Washing Area:</u>** The returned cylinders are conveyed through a washing area, where they are thoroughly washed and cleaned. This operation is mainly performed by a number of water jet sprays that are placed along the conveyor belt. This process is not considered in the simulation model. This is mainly due to the fact that its process time and breakdowns do not have any impact on the main cylinder filling operation.

- **<u>LPG Filling Process:</u>** The cylinders are pushed to filling stations to be filled with LPG. Each filling station is composed of four filling machines. Witness Single Machine object was used to model the filling machine operations. Also, using the Module object in the Witness simulator, every four filling machines were grouped into one Filling station.

A module is a discrete element that groups together a collection of other elements. This would allow modelling self-contained processes within a model. Figure 6-13 shows a filling station modelled as a module object and its contents, which are the filling machines.

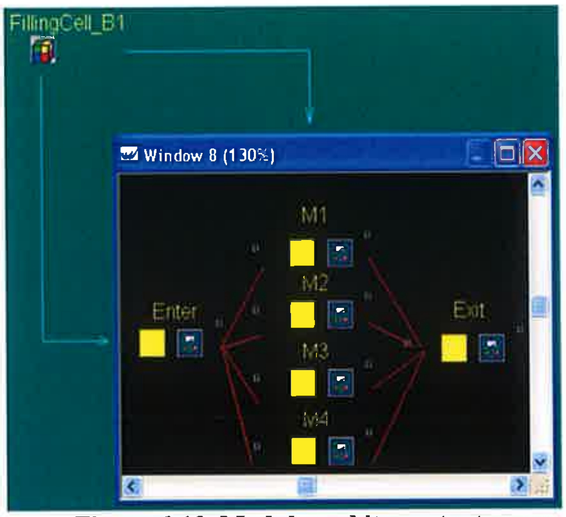


Figure 6-13, Module and its contents

Table 6-10 Inspection times for bottling plants

No.	BP	Inspection time (minutes)
1	Babol	0.090
2	Esfahan	0.045
3	Karaj	0.045
4	Kashan	0.135
5	Kerman	0.135
6	Mashhad	0.135
7	Qom	0.135
8	Yazd	0.270

Each Filling module has an Enter and Exit object. The LPG cylinders enter the module through the enter object. The cylinders are sent to filling machine 1 to 4. Once the cylinders are filled then, these are sent out of the module through the Exit object to the next stage in the cylinder process. The cycle time for filling cylinders are set to 65 seconds. Also, breakdown considerations for these machines incorporated in the model using the historical data available from the system. Appendix F provides such data.

- **Cylinder Inspection Area:** This process is modelled using a single machine object. Based on historical data only up to 5 % of the cylinders are rejected and the rest are moved to the dispatching area. This is supported using data provided in Appendix F. The inspection time is variable and it is based on the considering bottling plant in the system as indicated in the following table.

- **Cylinder Rework Area:** This operation is also modelled using a machine object. The cycle time for this process is established based on the historical data from bottling plants. Usually, 5% of the rejected cylinders are considered to have minor faults that are sent to the dispatching area. The remaining cylinders are directed to the second machine, which is a multi-cycle machine. This machine first removes the LPG contents of the cylinders to storage area. In the second cycle of this machine the cylinder maintenance is performed with a cycle time ranging from a minimum of 2 hours up to maximum time of 1 week. This cycle time is taken from the historical data provided in Appendix F.

- **Dispatching Area:** The fault free cylinders are shipped to the dispatching inventories to be taken by the company's agent for distribution between the end users. This area is modelled using a buffer object. The capacity of this buffer varies for each bottling plant in the system. Figure 6-14, shows a typical processing plant developed in the model.

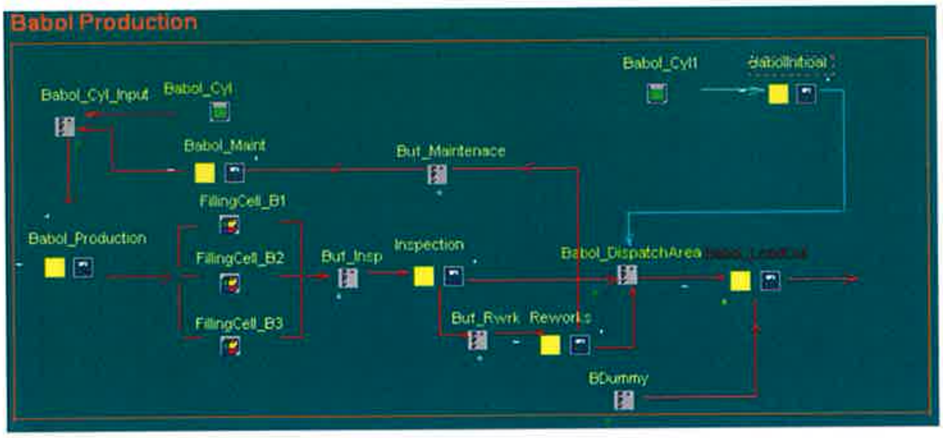


Figure 6-14 Simulation Model view showing input & output of a Bottling Plant

6.6 *Physical Supply of Raw Material*

The physical supply or distribution of raw material from refineries to bottling plants across the country is performed by means of Supper Trucks. Trucks and the drivers are one of the main resources used to perform the supply of raw materials to the bottling plants. As managing routing and scheduling of resources is the main aim of this project, for this reason the following section reviews how resources are modelled in Witness simulation environment.

6.6.1 Delivery Schedule

This simulation model was not established to develop delivery schedules but it was rather developed to examine and evaluate the effect of different delivery schedules on the overall system performance measures. In this approach, a monthly delivery schedule is externally generated using the genetic search engine. The generated schedules are stored in a database that is interfaced with the Witness simulation tool. Figure 6-15, illustrates the steps in generating schedules in the simulation model and transferring delivery data from the database to the simulation model for each truck.

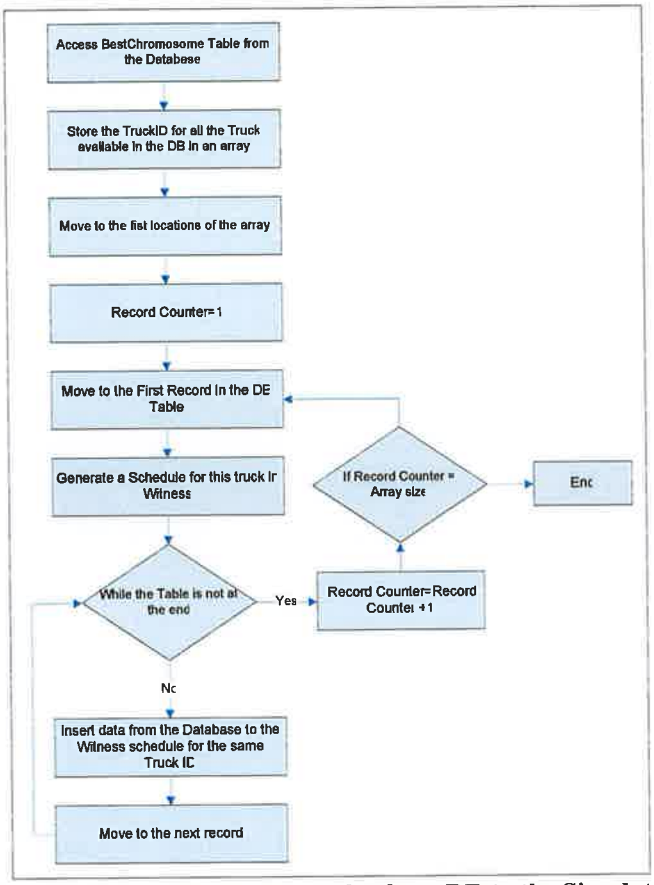


Figure 6-15 Transferring Schedules from DB to the Simulator

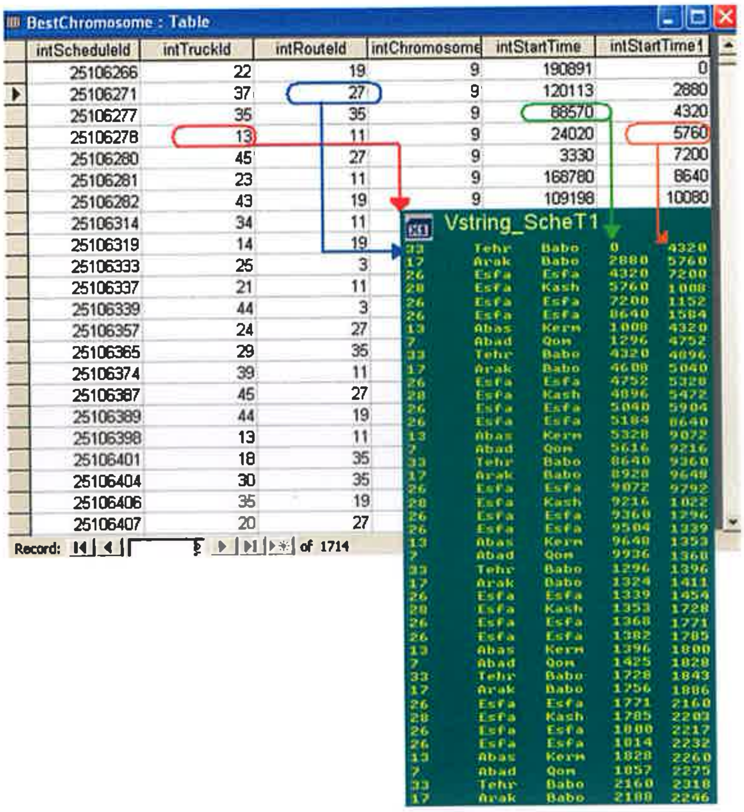


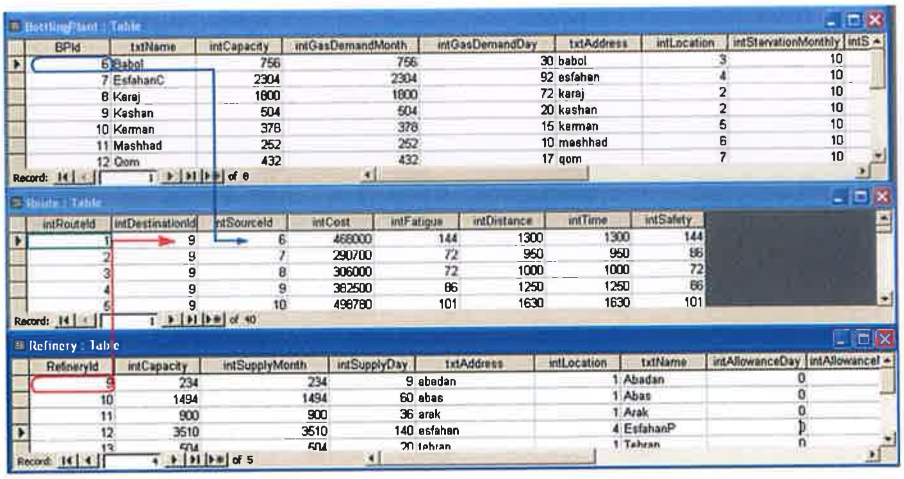
Figure 6-16 Schematic view showing data transfer to the simulation model

The number of transportation schedules generated in a simulation model depends on the total number of trucks that exists in a monthly schedule. For each truck existing in a schedule an equivalent schedule is generated in the simulation model. These schedules are string array variables with dimension of (5,100). Figure 6-16, shows an example of a string variable in Witness simulation environment that holds a transportation schedule for Truck1 in the model.

A raw in a schedule represents a route that a truck is travelling. Typically, a route starts with a start location (i.e. refineries) and an end location (i.e. bottling plants). In addition there is a start time assigned to each location in the route. The start time suggest the time to leave a location to move to the next location as specified in the scheduled.

6.6.2 Transportation Routes

The routes suggest the possible alternative paths that link refineries to bottling plants existing in the real system. In the considering supply chain, there are five refineries and eight bottling plants. Therefore, there are at least 40 possible routes linking refineries to bottling plants. It is important to mention that based on the size of the supply chain it is possible to extend the number of suppliers, plants and the routes between them. The possible routes existing in a system are defined earlier in Genetic search module. As illustrated in Figure 6-17, a route is stored in database using a route number, Refinery, Bottling plant, distance, speed limits and other parameters that are stored in the database. The user could add or delete routes from this database. As shown in Figure 6-18, these data are transferred to variables in the simulator for any further operations.



BottlingPlant : Table

BPId	txtName	intCapacity	intGasDemandMonth	intGasDemandDay	txtAddress	intLocation	intStarvationMonthly	intS
6	Babol	756	756	30	babol	3	10	
7	EsfahanC	2304	2304	92	esfahan	4	10	
8	Karaj	1800	1800	72	karaj	2	10	
9	Kashan	504	504	20	kashan	2	10	
10	Kerman	378	378	15	kerman	5	10	
11	Mashhad	252	252	10	mashhad	6	10	
12	Qom	432	432	17	qom	7	10	

Record: 1 of 8

Route : Table

intRouteId	intDestinationId	intSourceId	intCost	intFatigue	intDistance	intTime	intSafety
1	9	6	468000	144	1300	1300	144
2	9	7	290700	72	950	950	86
3	9	8	306000	72	1000	1000	72
4	9	9	382500	86	1250	1250	86
5	9	10	498780	101	1630	1630	101

Record: 1 of 40

Refinery : Table

RefineryId	intCapacity	intSupplyMonth	intSupplyDay	txtAddress	intLocation	txtName	intAllowanceDay	intAllowanceI
9	234	234	9	abadan	1	Abadan	0	
10	1494	1494	60	abas	1	Abas	0	
11	900	900	36	arak	1	Arak	0	
12	3510	3510	140	esfahan	4	EsfahanP	0	
13	504	504	20	tehran	1	Tehran	0	

Record: 1 of 5

Figure 6-17 Route Database Table

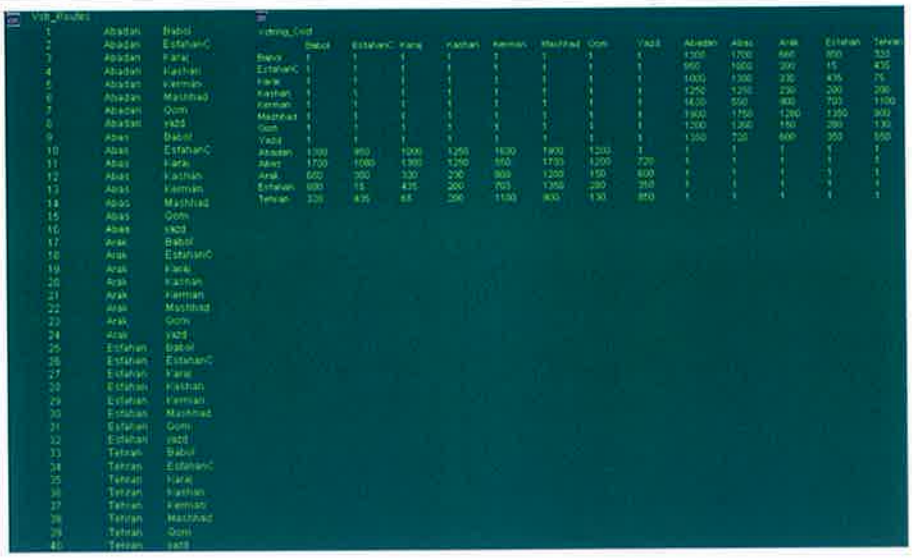


Figure 6-18 Data from route database transferred to Simulator

The time to travel these routes depends on the road type and weather conditions. Also, the loading status of the vehicle determines the maximum possible driving speed. In this approach historical data on the minimum, maximum and the average vehicle speed for each route were used to establish vehicle speed in routes. Refer to Appendix F for further information on these data. Table 6-11, suggests some speed parameters related to different routes exists in the system.

Table 6-11 Loaded/ Unloaded Vehicle speeds for a number of Transportation routes

	Loaded Speed(Km/hrs.)			Unloaded Speed(Km/hrs.)		
	Min	Mod	Max	Min	Mod	Max
1	50	75	100	100	125	150
2	55	80	105	100	125	150
3	55	80	105	100	125	150
4	55	80	105	100	125	150
5	55	80	105	100	125	150

6.6.3 Transportation Vehicle

The supper trucks could be modelled using either vehicle and track objects or the single machine object from the Witness object library. The use of vehicle and track objects had the advantage of better physical movements, which was useful for presentations. However, one of the main disadvantages of this approach was the lack of flexibility and complexity in modelling concepts such as breakdowns. As a result single machine objects were used to model supper trucks for transporting LPG. As illustrated in Figure 6-19, to model the trucks the following objects were used:

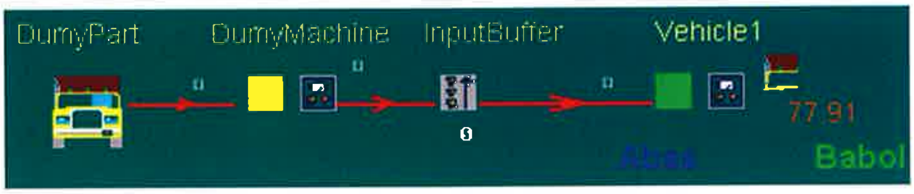


Figure 6-19 Transportation Vehicles

- **Dummy Part Object:** For each Truck used in a schedule a corresponding part object is generated in the simulation model. The parts are used to contain information about routes to be travelled by trucks in order to complete their monthly delivery schedules. There are a number of attributes assigned to the dummy parts once they are created in the model. These attributes are described in Table 6-12. These attributes are used to model the logic involved in LPG transportation.

Table 6-12 indicating the attributes assigned to dummy parts

No.	Attribute	Descriptions
1	**Nattr_CurrentLoc**	To hold the name of the current location of the truck
2	**Iattr_StageNumber**	The current position of the truck in routes to take
3	**Nattr_MachName**	The tucks name
4	**Iattr_RouteNo**	The routes are specified by numbers. They specify the path to travel by trucks.
5	**Nattr_LoadingDist**	The name of the Refinery to load from
6	**Rattr_LoadingST**	The start time to depart the refinery
7	**Nattr_UnloadingDist**	The name of the Bottling Plant to unload
8	**Rattr_UnloadingST**	The start time to depart from the bottling plants
9	**Iattr_Loaded**	To specify if the truck is loaded
10	**Iattr_TotalStages**	The total number of routes that must be taken
11	**Rattr_TimeLoaded**	To specify the loading time
12	**Rattr_TimeUnLoaded**	To specify the unloading time

The initial value for these attributes is transferred from the database to the corresponding part in the simulation model. As vehicles are moving between locations, these attributes are updated to reflect the current position and status of the truck with respect to the schedule.

- **Input Buffer:** Once the corresponding part is created in the simulation model, its attributes are initialised from the database table. The part is directed to wait in a buffer to be processed by a single machine, which is the actual truck. In fact, in this way the parts are the tasks to be performed by trucks and the buffer is used to list all tasks for the truck.

- **Single Machine object:** This object is mainly used to perform transportation of LPG from refineries to bottling plants. The cycle time for this machine is obtained based on the distance to travel, and the speed limit on the routes to travel. A truck's speed also depends on the loading condition of the truck.

Using the detail property page of this object one could model vehicle uncertain behaviours such as: breakdowns, repairs and scheduled maintenances. These are modelled using the historical data available from the trucks. Refer to Appendix F for details on these data.

When a part representing a task enters a machine object, it is checked to see if it is loaded or unloaded. Based on its loading status, its current location and its scheduled destination which could be either a refinery to load LPG from or a bottling plant to unload is recorded from the related attributes. In addition the route number to travel is used by a function to calculate its travelling speed. The vehicle speed is calculated based on a triangular distribution with its main parameters taken from the Logistics manager.

Once, the current location, destination and the vehicle speed are specified, the cycle time is calculated based on these parameters. From the current location and the destination, the travelling distance is calculated. Dividing this value by the vehicle speed the cycle time is obtained in minutes.

The total distance travelled by the truck is updated when the cycle time is elapsed. In addition, the total distance travelled based on the loading status of the truck is also calculated. This could be used to see the amount of deadhead travelling with respect to the truck schedule. Finally, the part is directed to the next location based on its destination attribute.

6.6.4 Part flow logic

The part flow logic refers to the raw material transportation, performed by trucks as described earlier. Trucks use monthly transportation schedules to carryout deliveries. The part flow logic would be clearer by considering and example. Assuming a truck is currently placed in location A. Based on the following transportation schedule shown in Table 6-13, the first route to be taken by the truck is route 33. Once a part is generated in the simulation model, the main part attributes are updated as indicated in Table 6-14.

Table 6-13 Transportation Schedule

Route No.	Refinery	Start Time (Min)	Bottling Plant	Start Time (Min)
33	Tehran	0	Babol	440
17	Arak	1880	Babol	2680
26	Esfahan	4120	Esfahan	4255

Table 6-14 Truck's main attributes

No.	Attribute	Descriptions
1	Nattr_CurrentLoc	A
2	Iattr_StageNumber	1
3	Nattr_MachName	Truck1
4	Iattr_RouteNo	33
5	Nattr_LoadingDist	Tehran
6	Rattr_LoadingST	0
7	Nattr_UnloadingDist	Babol
8	Nattr_UnloadingDist	440
9	Iattr_Loaded	0 (i.e. the truck is not loaded yet)
10	Iattr_TotalStages	3
11	Rattr_TimeLoaded	To be specified when the truck is loaded
12	Rattr_TimeUnLoaded	To be specified when the truck is un loaded

In the simulation model a part with the above information is generated and it is send to a machine object representing a truck. The truck would be busy with a cycle time equal to the travel time from Location A to loading destination called "Tehran", where the refinery is located.

Upon the arrival to the refinery, the truck would be waiting in a queue for LPG filling operation. Once the filling is performed the tuck is then transferred to another buffer. In this outgoing buffer, trucks are queuing based on their Departure time. This means that the truck with the minimum Departure time (i.e. Rattr_LoadingST) would be shipped out the refinery.

Using the unloading destination attribute (Nattr_UnloadingDist) trucks would be kept busy for the duration of travel time from the loading refinery to the unloading bottling plant. Once this cycle time is elapsed, trucks would be waiting in a queue for unloading at the plant. It is important to mention that the travel time also includes the rest time that drivers may take on travelling a particular route.

Once, the LPG unloading operation is performed, trucks are then transferred to another buffer, Where they would be waiting until their departure times (i.e. Nattr_UnloadingDist) reach the simulation time.

If trucks departure time were already over, then they would be dispatch as soon as arriving into the waiting buffer. At this stage trucks are moved to the next stage of their schedules. As a result the stage numbers are incremented by one (i.e. Iattr_StageNumber) and the main attributes of trucks are updated as shown in the following table.

Table 6-15 Updated attributes

No.	Attribute	Descriptions
1	**Nattr_CurrentLoc**	Babol
2	**Iattr_StageNumber**	2
3	**Nattr_MachName**	Truck1
4	**Iattr_RouteNo**	17
5	**Nattr_LoadingDist**	Arak
6	**Rattr_LoadingST**	1880
7	**Nattr_UnloadingDist**	Babol
8	**Nattr_UnloadingDist**	2680
9	**Iattr_Loaded**	0 (i.e. the truck is not loaded yet)
10	**Iattr_TotalStages**	3
11	**Rattr_TimeLoaded**	To be specified when the truck is loaded
12	**Rattr_TimeUnLoaded**	To be specified when the truck is un loaded

The above steps are continued until the stage number exceeds the total stages to be travelled by trucks. When this happens, trucks are shipped out of the simulation model ending their LPG delivery operations.

6.7 Modelling dynamic behaviours of systems

As mentioned earlier there are a number of sources of variability within the supply system. These are mainly due to unscheduled breakdowns and repair of equipments within the supply ports, transportation fleet and bottling plants. Mainly the breakdown/repair models for these equipments are established based on the maintenance and daily production records. The maintenance records from supply sources, trucks and bottling plants provided data on machine breakdowns that required the use of maintenance personnel for repairs. Mostly data were collected for a year period and analysed. The distribution fitting software BestFit from Palisade Corporation [157] was used to generate the best distribution for Mean Time Between Failures (MTBF) and Mean Time To Repair (MTTR) for each machine.

Also, analysing the daily production records suggested breakdown patters that are operation dependent failures. There is no need of maintenance personnel for these repairs. For example a common cause of down time on filling stations was lack of oil lubrication problem. Also, this pattern was detected for trucks transporting LPG to bottling plants. Data was collected based on a year period and the BestFit was used to generate distributions for mean number of operations to next failure and mean time to repair. The breakdown-repair models can be developed in Witness using the Breakdown sheet of the machine description dialogue box as shown below.

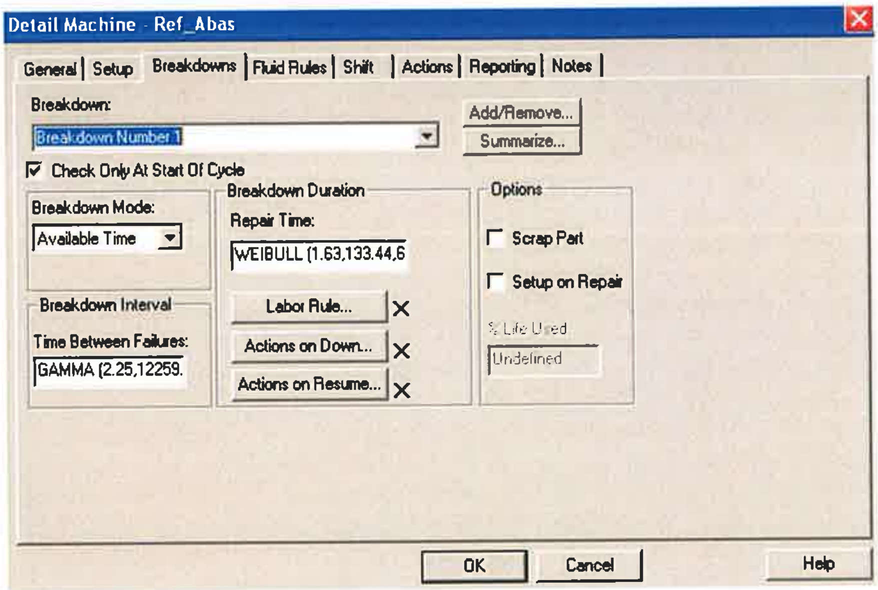


Figure 6-20 Breakdown Menu Options

Also, daily bottling plant consumption of LPG is seasonal and variable. To model this variability in LPG demand data were collected for each month for the last three years. Once again the BestFit was used to establish the best distribution curve modelling monthly demand for LPG for each bottling plant considered in the system.

6.8 Model Validation

There are three important issues that must be addressed by the simulation analyst, known as model verification, validation and establishing credibility in the model. As suggested by Law and Kelton [121], the model validation and verification is probably the single most difficult task that a simulation analyst must face. If a model is not valid or cannot be proven to be valid then the validity of inferences and conclusions derived from the model output must also be in doubt. Verification is concerned with proving that the conceptual simulation model has been translated correctly into the computer simulation model. In the other hand, validation is concerned with determining whether or not the conceptual model is an accurate representation of the system being studied.

Model credibility is established once management accepts the model as being valid and uses the model as a decision support tool. One area contributing to credibility of the model is the animation of simulation outputs. Animation is seen as an effective method for communicating the basic model construct to managers who were not mainly involved in model development. [148] suggest that ongoing model development in which the need for verification and validation causes continual comparison of the real system to the conceptual model and to the operational model and repeated modification of the model to improve its accuracy. To this effect [121] present validation and verification stages as shown in Figure 6-21.

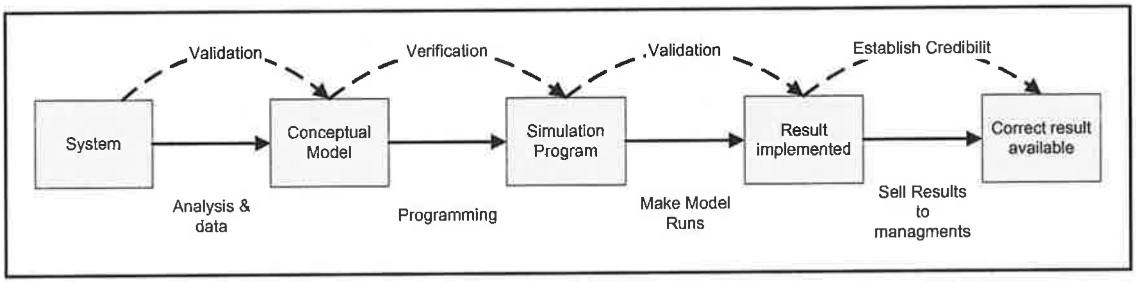


Figure 6-21, Validation, verification and establishing credibility [121]

6.9 Establishing a Valid and Credible Model

In this section the steps taken to establish a valid and credible model are reviewed. [121] present a modified three step approach to model validation based on the Naylor and Finger [158] and [159]. The approach does not guarantee an absolutely valid model but it will result in a model that is more representative of the real system and also more credible.

- ***STEP 1: Develop a model with high face validity:*** This step is to ensure that the model developed seems reasonable to people who are familiar with the system under study. To achieve this, a model builder should examine and incorporate information from a variety of distinct sources. In the case of the company involved in this project the model was developed by constant communication with individual experts in the company. Therefore, extensive use was made from expert's knowledge familiar with the system.

As the system under study already exists, data were obtained from the real system and incorporated into the model.

- ***STEP 2: Test the assumption of the model empirically:*** the aim of this step is to test quantitatively the assumptions of the model. Tools and techniques such as graphical plots and goodness of fit tests can assess the adequacy of fitted distributions. Also sensitivity analysis can be employed during this step to determine whether or not the model output is sensitive to changes to the value or properties of a factor of interest. If the output is found to be sensitive to some aspect, then that aspect must be modelled carefully and in detail. In some cases there might be a need to determine the sensitivity of the model to two or more factors and their possible interactions. This can be done by employing statistical experimentation design techniques.
- ***STEP 3: Determine how representative the simulation output data are:*** As [121] suggested the most definitive test in establishing the validity of a simulation model is that its output data closely resembles the output data that would be expected from the actual system. If the system exists, as in this case, then the output data of the simulation model are compared to those of the actual system, the result of this comparison determined the validity of the model. If this comparison is found to be favourable then the model is said to be valid, otherwise the model is modified until it is.

This however, raises the question of whether a model that has been modified until a particular set of input data is validated or merely representative of the particular set of input data. It is recommended using one set of data to calibrate the model and another to validate the model.

Besides the statistical approaches, there are other non-statistical based approaches for model validations. An example of such techniques is called Turing test. In the Turing test system experts are firstly asked to examine one or more sets of system data and one or more sets of model data without knowing, which is which. In the case of correct distinctions, they are asked to provide and explanation for their justifications. Based on the explanations, the model must be improved. Also, the system experts could be used to review the model data for reasonableness. Throughout this project the model and data were shown to the experts several times and this aided immensely in establishing a credible model.

6.10 *Statistical procedure for model validation*

The test of validity reduces down to comparing the results generated by the simulation model to the data collected from the real system operating under the same conditions. A number of statistical tests have been suggested over the years for the purpose of model validation. As suggested by [121], there are three approaches that have been used widely by practitioners, namely known as the inspection, confidence interval and time-series approaches. The first two approaches are introduced in the following sections.

6.10.1 The inspection approach

This is perhaps one of the mostly used approaches by simulation developers. In this approach the statistical data is collected on one or more statistics from the real world system. The same statistical data is collected from the simulation model. Then a comparison is carried out on these data without using any formal statistical procedures examples of statistics that might be used for this approach are sample mean, sample variance and histograms.

[121] state that there is an inherent difficulty with this approach in that "each statistic is essentially a sample size of 1 from some underlying population, making this idea particularly vulnerable to the inherent randomness of the observations from both the real system and the simulation model". The ***correlated inspection*** approach was then recommended to overcome this problem. In this approach, the model is derived using historical system input data rather than samples from distributions and then the comparison are made. This approach is particularly useful for situations where there is a limitation on the amount of data that may be collected from either the real system or the model.

6.10.2 Confidence interval approach

In situation where it is possible to collect large amounts of data from both the real system and the model this approach is a much more reliable approach than the inspection approach. This approach requires that ***m*** independent sets of data to be collected from the real system and ***n*** sets of data from the model. Then the χ_j be the average of the observations in the j^{th} set of system data and Y_j be the average of the observations in the j^{th} set of model data. The χ_j's are **I**ndependent **I**dentically **D**istributed **(IID)** random variables with mean $\mu_x = \mathrm{E}(\mathrm{Y}_j)$ and Y_j's are IID random variables with mean $\mu_y = \mathrm{E}(\mathrm{Y}_j)$. The confidence interval approach attempts to compare the model with the real system by constructing a confidence interval for $\zeta = \mu_x - \mu_y$.

In order to construct the confidence interval, either of the two approaches can be used, namely the ***paired t-test*** [121] approach and ***Welch's*** approach [160]. The ***paired t-test*** approach required that $\mathrm{M} = \mathrm{N}$ but allows χ_j to be correlated with Y_j. The ***Welch*** approach does not require that $\mathrm{M} = \mathrm{N}$ but requires that the χ_j's are independent of the Y_j's. The approach may be used for any values of $\mathrm{M} \geq 2$ and $\mathrm{N} \geq 2$.

6.11 Validation Results

This section is primarily concerned with the model validation. To determine how representative of the real system the simulation output data are, the confidence-interval approach based on independent data was proposed. For this reason the ***Welch*** approach was conducted as the ***M*** is not equal to ***N*** and there is no intention to truncate one of the data set or to enlarge the other. As required by this approach the data sets are independent.

The main two indicators used to validate this model were based on average daily Inventory levels and the average LPG cylinders filled daily (i.e. outputs) at different bottling plants. The company records and reports were used to collect and document all relevant data. The existing inventory level at different bottling plants were collected for a period of time form the 1st of March 2004 to 1st of December 2004 from Inventory loading report forms that operators are required to fill out and maintain.

6.11.1 Result of the validation procedure

As specified earlier, when comparing two systems with unequal and unknown variances one of the best approach is the Welch method. In our case the sample sizes are different. In the case of inventory level validation there are only five sets of independent data available from the real system. Also to validate the demand level, there are only three sets of data available from the real system. However, it is possible to generate as many sets of independent data from the model as desired. The following table represents the average monthly inventory level for each bottling plant in the considering supply chain. These data are collected from simulation model for Transportation schedule. The data sets are from ten independent replications from the simulation model. Each replication uses a different seed number.

Table 6-16 Average Monthly Inventory Levels of Bottling plants from simulation model

No. Bottling Plant / Reps (n)	1 Babol Avg. Volume(tonnes)	2 Esfahan Avg. Volume(tonnes)	3 Karaj Avg. Volume(tonnes)	4 Kashan Avg. Volume(tonnes)	5 Kerman Avg. Volume(tonnes)	6 Qom Avg. Volume(tonnes)	7 Mashahad Avg. Volume(tonnes)	8 Yazd Avg. Volume(tonnes)
1	51.11	72.59	277.87	28.93	66.55	5.03	38.00	15.62
2	61.20	67.99	287.97	29.02	60.81	23.33	37.43	10.75
3	39.39	55.16	291.02	25.65	56.81	8.46	40.77	13.54
4	85.03	76.93	301.51	31.64	60.93	3.02	42.98	13.42
5	72.44	87.04	314.66	26.40	73.04	6.57	42.53	13.17
6	66.94	67.72	287.99	20.82	75.14	15.01	41.68	15.81
7	60.21	71.49	292.83	24.21	74.28	10.99	39.87	14.16
8	58.22	54.69	275.09	30.62	71.51	7.28	37.50	11.88
9	65.37	55.99	319.61	21.28	58.85	8.17	39.89	12.79
10	71.34	84.86	296.40	29.41	79.94	26.66	40.51	14.59

Similarly, average monthly inventory levels are collected from the real system as illustrated in the following table. There are only five sets of data available from the system due to the difficulties in gathering data from the real system.

Table 6-17 Average Inventory levels for Bottling Plants from the real system

No. Bottling Plant / Set (n)	1 Babol Avg. Volume(tonnes)	2 Esfahan Avg. Volume(tonnes)	3 Karaj Avg. Volume(tonnes)	4 Kashan Avg. Volume(tonnes)	5 Kerman Avg. Volume(tonnes)	6 Qom Avg. Volume(tonnes)	7 Mashahad Avg. Volume(tonnes)	8 Yazd Avg. Volume(tonnes)
1.00	84.31	69.97	292.31	32.78	82.94	11.66	38.63	18.38
2.00	62.09	95.72	310.75	27.38	46.88	11.78	41.50	12.94
3.00	52.06	71.16	266.41	28.56	68.47	10.44	39.47	13.22
4.00	65.09	68.34	290.25	32.09	68.41	32.84	41.72	13.88
5.00	63.16	52.84	283.50	27.00	67.91	11.19	41.84	14.53

The Welch method can be initiated with calculating the mean and standard deviation of both the real system and the simulation model. It is then possible to obtain the estimated degree of freedom. This calculation is done using the following equation:

$$\hat{f} = \frac{\left[\left(S_1^2(n_1)/n_1\right) + \left(S_2^2(n_2)/n_{21}\right)\right]^2}{\dfrac{\left(S_1^2(n_1)/n_1\right)^2}{(n_1 - 1)} + \dfrac{\left(S_2^2(n_2)/n_{21}\right)^2}{(n_2 - 1)}} \quad \text{.............................} (74)$$

Where S_1^2 and S_2^2 are the standard deviation of the simulation model and the real system respectively and n_1 and n_2 are the sample sizes. Having obtained the degree of freedom the following equation is used to calculate the confidence intervals. Table 6-18, summarises the result of computation for confidence intervals at different levels.

$$\bar{X}_1(n_1)-\bar{X}_2(n_2)\pm t_{\hat{f},1-\alpha/2}\sqrt{\frac{S_1^2(n_1)}{n_1}+\frac{S_2^2(n_2)}{n_2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (75)$$

Table 6-18 Result of Welch Confidence Interval for Inventory Levels

Confidence Intervals	1 Babol		2 Esfahan		3 Karaj		4 Kashan	
	Lower Interval	Upper Interval	Lower Interval	Upper Interval	Lower Interval	Upper Interval	Lower Interval	Upper Interval
AT 99%	-16.31	20.74	-22.38	26.70	-31.25	19.54	-1.85	7.38
AT 98%	-13.53	17.96	-18.23	22.55	-27.17	15.47	-1.19	6.72
AT 95%	-9.82	14.25	-11.99	16.31	-21.07	9.36	-0.29	5.82

Confidence Intervals	5 Kerman		6 Qom		7 Mashahad		8 Yazd	
	Lower Interval	Upper Interval	Lower Interval	Upper Interval	Lower Interval	Upper Interval	Lower Interval	Upper Interval
AT 99%	-20.67	18.93	-10.83	19.08	-1.97	3.00	-2.45	4.49
AT 98%	-17.32	15.58	-8.43	16.68	-1.61	2.64	-1.87	3.90
AT 95%	-13.56	11.82	-5.32	13.58	-1.13	2.16	-0.94	2.97

As it can be observed from Table 6-18, at all chosen levels, zero is contained with in the intervals. Hence one can conclude that statistically the difference between the simulation model and the real system is insignificant and therefore the simulation model is a valid representation of the real system.

The second parameter considered to validate the simulation model was Cylinder output from each bottling plant existing in the system. For this reason the average monthly cylinder outputs were collected from the simulation model. In total ten data sets were gathered from ten different simulation runs, each having different seed values.

Table 6-19, Average Monthly LPG Cylinder outputs for Bottling Plants from Simulation Model

No.	1	2	3	4	5	6	7	8
Bottling Plant	Babol	Esfahan	Karaj	Kashan	Kerman	Qom	Mashahad	Yazd
Reps (n)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)
1	28.09	71.93	75.44	19.60	14.28	16.10	18.25	19.50
2	28.44	75.96	72.57	18.93	15.17	13.79	18.64	20.31
3	29.93	81.99	70.37	19.72	16.00	14.41	18.10	19.97
4	24.96	72.08	67.67	19.27	15.77	15.70	18.25	19.81
5	27.84	71.43	66.03	20.28	14.97	14.80	17.99	19.37
6	27.75	78.04	70.85	20.97	13.40	14.11	17.99	19.56
7	30.36	74.61	69.65	19.62	12.99	15.29	18.50	19.96
8	29.46	79.58	72.31	18.56	13.93	14.88	18.09	19.89
9	28.29	81.82	63.64	20.70	16.11	14.62	18.18	20.02
10	28.04	75.88	67.41	20.20	14.01	14.27	18.03	20.16

Also, the corresponding data were gathered from the real system. In total three data sets were gathered from the real system as shown in Table 6-21.

Table 6-20 Average Monthly LPG Cylinder outputs from Bottling Plants from the real system

No.	1	2	3	4	5	6	7	8
Bottling Plant	Babol	Esfahan	Karaj	Kashan	Kerman	Qom	Mashahad	Yazd
Set (n)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)	Avg. Cylinder output (Tonnes)
1	28.09	74.53	65.81	21.25	15.16	13.25	18.34	20.31
2	31.78	75.84	70.35	19.31	13.44	17.06	19.97	21.38
3	30.59	82.16	73.00	19.56	16.31	14.72	17.19	19.25

Conduction, the Welch method at different confidence intervals resulted in the lower and upper bond limits as indicated in Table 6-21. Observing these intervals suggest zero is contained within the corresponding limits and therefore the developed model and the real system is statistically indifferent and therefore this adds to the validity and credibility of the developed model.

Table 6-21, Result of Welch Confidence Interval for LPG Cylinder Demands

Confidence Intervals	1		2		3		4	
	Babol		Esfahan		Karaj		Kashan	
	Lower Interval	Upper Interval	Lower Interval	Upper Interval	Lower Interval	Upper Interval	Lower Interval	Upper Interval
AT 99%	-3.54	7.22	-10.92	13.28	-10.64	10.89	-2.72	3.23
AT 98%	-2.28	5.97	-8.10	10.46	-8.12	8.38	-2.02	2.54
AT 95%	-0.95	4.63	-5.09	7.45	-5.45	5.71	-1.28	1.80

Confidence Intervals	5		6		7		8	
	Kerman		Qom		Mashahad		Yazd	
	Lower Interval	Upper Interval	Lower Interval	Upper Interval	Lower Interval	Upper Interval	Lower Interval	Upper Interval
AT 99%	-3.80	4.42	-7.68	8.11	-3.71	5.58	-2.63	3.55
AT 98%	-2.85	3.46	-5.28	5.71	-2.30	4.17	-1.69	2.61
AT 95%	-1.82	2.44	-3.10	3.52	-1.01	2.88	-0.38	1.29

6.12 Conclusions

This chapter initiated with a brief description on activities involved in a typical supply chain. To this effect, different elements comprising the considering LPG supply chain was described. Based on the cost analysis performed, it was found that the supply of raw material from the supply ports to the processing plants would take the highest ratio of the actual physical transportation cost in this company. To this effect addressing complex issues in vehicle routing and scheduling for LPG transportation could greatly assist management with this task and ultimately leading to cost reduction for the company.

The considering system is subjected to high variability due to the low degree of automation in the system and also high uncertainty due to unpredictable influence of environment, human, operations and market. Obviously, traditional methods of studying supply chain/ manufacturing systems such as linear programming or flow charting are incapable of solving the complex nature of this vehicle routing and scheduling.

Simulation modelling provides a better way to capture systems dynamics and therefore a better approach for modelling the supply chain. However, development of a simulation model is a difficult and complex task that requires the use of knowledge and expertise from many individuals within the system.

The computer simulation of the LPG supply chain system was validated using the Welch method and it was shown that the output of the model is representative of the expected output from the real system.

This model can now be used to aid in the LPG transportation decision-making process. This module provides a better-detailed model to further evaluate alternative transportation schedules provided by the search engine. The model will only remain valid as long as it is updated with current system data. For instance changes in transportation and processing times must be updated in the model. Finally, the simulation model can also be used as a stand-alone tool to evaluate different hypothesis and conduct different experiments on the considering supply chain.

Chapter 7: Results and Evaluations

7.1 Introduction:

The primary aim of this chapter is to show the applicability and effectiveness of the proposed hybrid genetic based decision support tool in addressing multi-objective vehicle routing and scheduling problems. This chapter initiates by comparing the developed Genetic algorithm with other methods to validate and evaluate the performance of this approach. Moreover, experiments performed to further assess the impact of population size, selection methods, crossover, mutation operators and the elitism on the GA performance are provided in this chapter. Lastly, the simulation model from the considering supply chain was used to evaluate the generated optimum solutions from the genetic search. The results from these evaluations are presented and the feedbacks from the simulation module to the search engine are discussed. Finally, an overall discussion of results and final conclusions drawn on this hybrid decision support tool for addressing multi-objective logistics problems are presented.

7.2 Genetic algorithms Validation

The aim of validation of genetic search engine here is to show that this method is searching for the global minimum. For this reason this method is compared with other approaches to establish its validity. In comparing this method, one can also evaluate the performance of this method. To this effect the Genetic search method is compared with TORA a linear transportation optimisation method and random search to find the optimum transportation cost. Further to this comparison, GA is compared with random search to find the optimum Vehicle Routing and Scheduling cost. These scenarios are further described as follows:

7.2.1 Optimum Transportation cost using TORA

The first comparison is based on the linear formulation of transportation problem. Evidently, the considering problem does not represent a linear relation between objective functions and primary factors in this model. However, the problem can be tailored to a simple transportation problem by just considering supply and demand levels. Also the distances could be considered as the cost measure to transfer the required LPG to the demand points. In this way, the only objective is to minimise travelled distance. For this purpose, distance cost is taken to be in linear relation with the travelled distance. This problem can be formulates as there are five refineries with LPG reserves and also, there are eight processing plants with the specified demand levels as shown in Table 7-1. As indicated from this table the supply and demand levels are considered to be equal. The distance travelled to link refineries to processing plants are presented in Table 7-2. The aim is to minimise the distance travelled to supply all the requested demands for LPG.

Table 7-1 Supply and Demand levels

Processing Plants	LPG Demands (Tonnes)
Babol	1148
EsfahanC	1881
Karaj	2500
Kashan	500
Kerman	539
Mashhad	824
Qom	535
Yazd	573
Total	8500

Refineries	LPG Reserves (Tonnes)
Abadan	915
Abas	2812
Arak	300
Esfahan	2983
Tehran	1490
Total	8500

Table 7-2 Distances linking sources and demands points

	Refinery	Processing Plant	intDistance	Route No.	Refinery	Processing Plant	intDistance
1	Abadan	Babol	1300	21	Arak	Kerman	900
2	Abadan	EsfahanC	950	22	Arak	Mashhad	1280
3	Abadan	Karaj	1000	23	Arak	Qom	150
4	Abadan	Kashan	1250	24	Arak	Yazd	600
5	Abadan	Kerman	1630	25	Esfahan	Babol	800
6	Abadan	Mashhad	1900	26	Esfahan	EsfahanC	15
7	Abadan	Qom	1200	27	Esfahan	Karaj	435
8	Abadan	Yazd	1350	28	Esfahan	Kashan	200
9	Abas	Babol	1700	29	Esfahan	Kerman	703
10	Abas	EsfahanC	1080	30	Esfahan	Mashhad	1350
11	Abas	Karaj	1380	31	Esfahan	Qom	280
12	Abas	Kashan	1250	32	Esfahan	Yazd	350
13	Abas	Kerman	550	33	Tehran	Babol	320
14	Abas	Mashhad	1750	34	Tehran	EsfahanC	435
15	Abas	Qom	1200	35	Tehran	Karaj	65
16	Abas	Yazd	720	36	Tehran	Kashan	200
17	Arak	Babol	680	37	Tehran	Kerman	1100
18	Arak	EsfahanC	300	38	Tehran	Mashhad	900
19	Arak	Karaj	330	39	Tehran	Qom	130
20	Arak	Kashan	230	40	Tehran	Yazd	850

The above problem was addressed, using the well-known transportation model TORA Optimisation System - Version 1.044, developed by Taha [161]. The total transportation cost obtained as 280959.17, which is distance travelled in Km as indicated in Table 7-3.

Table 7-3 Minimum distance travelled using TORA

From	To	Amount	Unitcost	Route cost	Number of Trips (Amount/18)	Trip Cost
S1	D1	0	1300	0		
	D2	0	950	0		
	D3	915	1000	915000	50.83	50833.33
	D4	0	1250	0		
	D5	0	1630	0		
	D6	0	1900	0		
	D7	0	1200	0		
	D8	0	1350	0		
S2	D1	0	1700	0		
	D2	0	1080	0		
	D3	641	1380	884580	35.61	49143.33
	D4	0	1250	0		
	D5	539	550	296450	29.94	16469.44
	D6	824	1750	1442000	45.78	80111.11
	D7	235	1200	282000	13.06	15666.67
	D8	573	720	412560	31.83	22920.00
S3	D1	0	680	0		
	D2	0	300	0		
	D3	0	330	0		
	D4	0	230	0		
	D5	0	900	0		
	D6	0	1280	0		
	D7	300	150	45000	16.67	2500.00
	D8	0	600	0		
S4	D1	0	800	0		
	D2	1881	15	28215	104.50	1567.50
	D3	602	435	261870	33.44	14548.33
	D4	500	200	100000	27.78	5555.56
	D5	0	703	0		
	D6	0	1350	0		
	D7	0	280	0		
	D8	0	350	0		
S5	D1	1148	320	367360	63.78	20408.89
	D2	0	435	0		
	D3	342	65	22230	19.00	1235.00
	D4	0	200	0		
	D5	0	110	0		
	D6	0	900	0		
	D7	0	130	0		
	D8	0	850	0		
					Total Transporation	280959.17

It is important to mention that the purposed GA search method was not primarily designed based on the general transportation formulation. It was designed to find near optimum transportation schedule considering different objectives and constraints in the system. Therefore the chromosome representation is based on random assignments of trucks, routes and start times. In the other hand TORA approach is single linear optimisation method that does not consider any real system constraints or resource capacity restrictions and therefore comparing the results with GA method could be misleading, if failing to consider the existing differences.

7.2.2 Optimum Transportation cost using Random Search Method

Similarly, a random search was performed to find a schedule with minimum transportation cost. In this approach parameters used in the transportation schedules are randomly chosen from their respective decision space. As shown in Figure 7-1, almost 20000 schedules are randomly generated. The schedules with the maximum and minimum transportation costs are 412,174 Km and 372,178 Km respectively. The minimum distance was obtained after 17,626 generations.

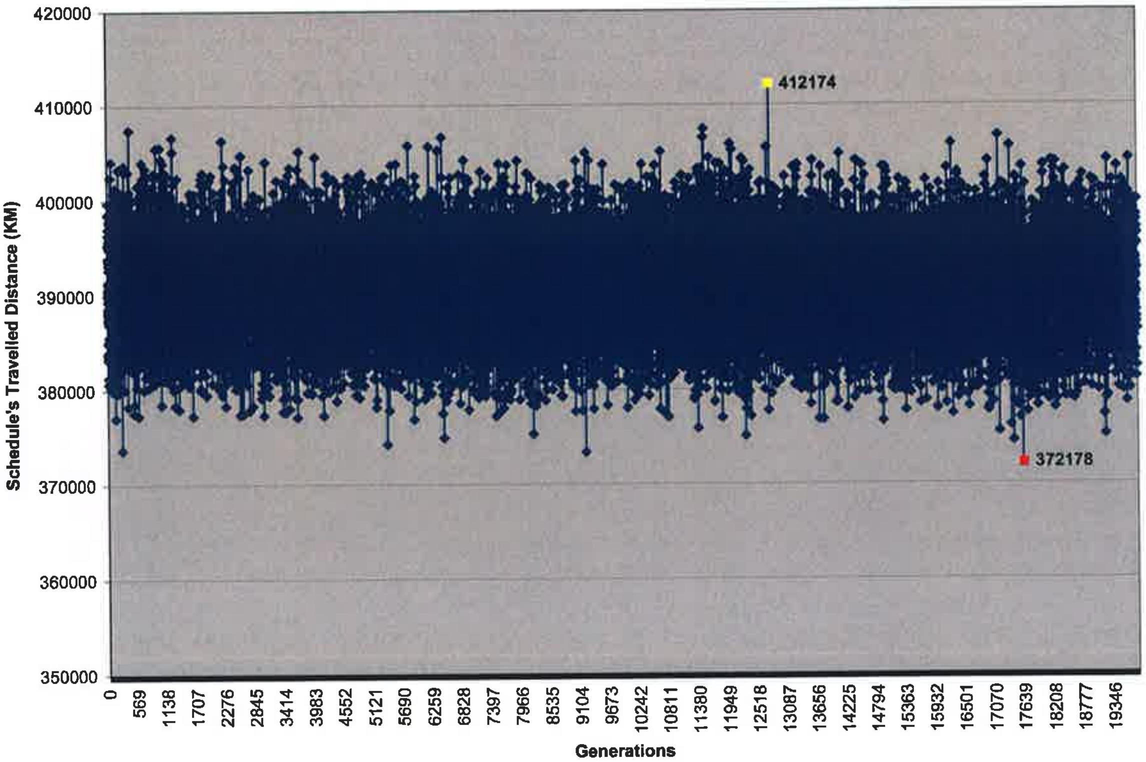


Figure 7-1 Random search to find the minimum travelled Distance

7.2.3 Optimum Transportation cost using Genetic algorithm Method

The proposed Genetic Search Method was used to find the minimum transportation cost. For this reason, the initial conditions for the GA search were set to *Pop Size* = 10, *Pc* = 80%, *Pm* = 20%, *Ngen* = 3500. As shown in Figure 7-2, the minimum transportation cost was 292,000 Km. This is almost 80178 Km less than the best-found solution in random search. Also, GA reached this solution after 950 generations.

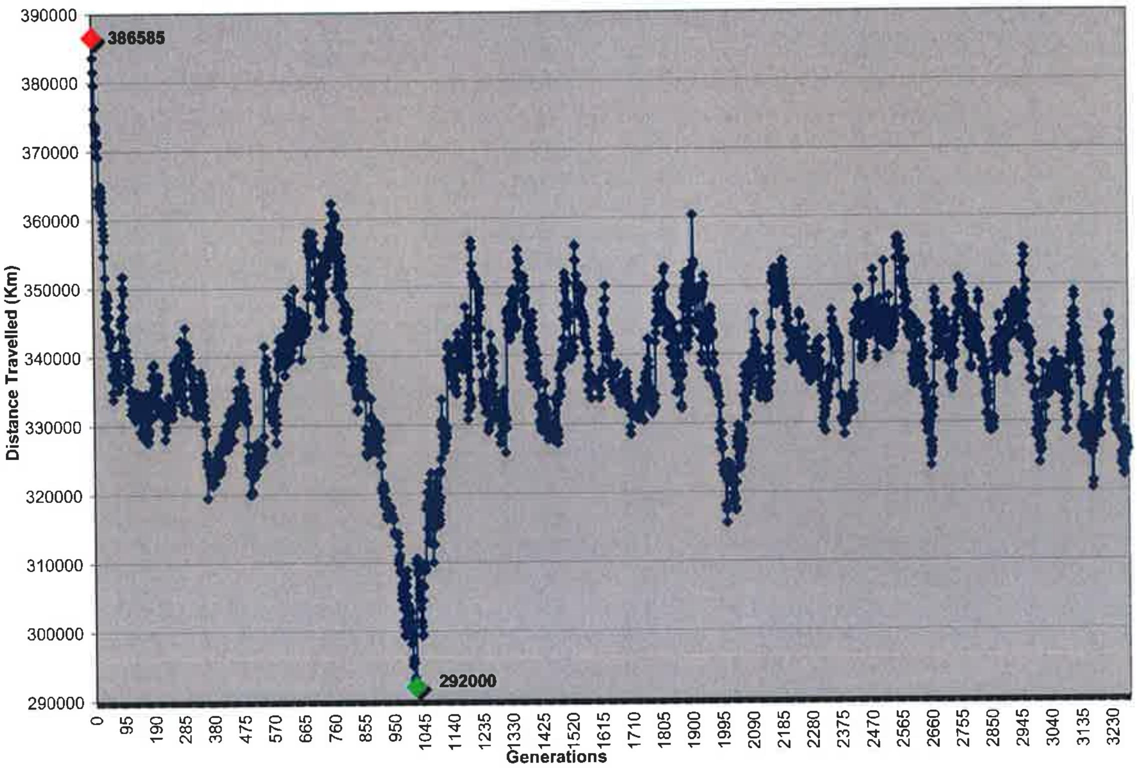


Figure 7-2 Minimum travelled distance using GA search Method

7.2.4 Optimum Vehicle Routing and Scheduling Cost

The above sections tried to show the GA performance in finding the minimum travelling distance. Here the aim is to show how the GA performs in finding the optimum vehicle routing and scheduling cost, which this method was initially designed for. To this effect the GA performance is compared with random search method.

Accordingly, a random search was conducted to find the minimum Vehicle Routing and Scheduling cost, considering all the possible costs in the schedule. Setting this method to run for 20,000 generations, the minimum cost obtained was about 21,763,302.23. This solution was reached after 10,000 generations.

Similarly, GA was used to find the minimum cost. The initial conditions for the GA search were set to *Pop Size* = 10, *Pc* = 80%, *Pm* = 20%, *Ngen* = 20,000. Figure 7-3, illustrates the minimum schedule cost obtained using these parameters. The minimum cost was 17,657,026.54. This schedule was obtained after 2649 generations. The obtained schedule is 4,106,275.69 units less than the minimum found solution using the random search.

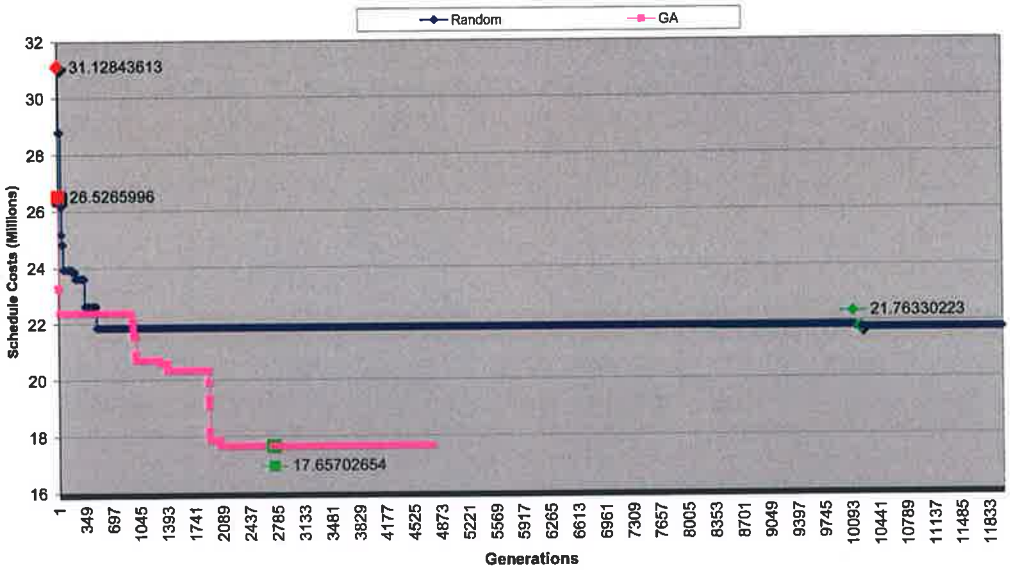


Figure 7-3 Minimum Schedule Cost obtained using Random & GA Search methods

7.3 Optimum values for Initial GA parameters

The described GA needs the tuning of several parameters. The main parameters are as follows:

1. ***Population Size (Pop):*** this value indicates the number of individuals in the population, which usually depends on the problem to be solved. [162] suggests using small sized population for problems where the initial population consists of feasible solutions. As finding a feasible solution might be as difficult as finding the optimal solution, it is also suggested to consider small size of populations, according to the computation time needed to obtain such a solution. Also, in the case where the initial population is not made of feasible solutions, it seems better to consider large populations. In this approach, repair techniques are used to address any infeasible solution within the population. Therefore using small population size is recommended.
2. ***Number of generations (Ngen):*** The current population is called generation in GA. This is one of the main stopping criterions used in GA applications. The number of generations ***(Ngen)*** is typically fixed in the beginning of the GA run. It is usually suggested to have largest size possible for ***Ngen*** considering the trade off between the computation time and ***Ngen***.
3. ***Crossover Probability (Pc):*** This parameter is usually used to decide which couples of individuals constituting the current population to be crossed. In this application, the

number of crossover operations (N_c) depends on the crossover probability P_c, which is specified by the user. Once, the parent chromosomes are selected, a random number R_c is generated. If $R_c<P_c$ then the operations occurs. Also the location of the crossover is calculated using: L_c =*INT [n-1] P_{c2}+1]*, where n is the length of the chromosome and P_{c2} is a random number ranging between [0,1]. ,

4. ***Mutation probability (Pm):*** this event is a rare phenomenon and occurs with a small probability. As in the case of the crossover operation, the number of mutation operations, N_m, depends on the mutation probability P_m, which is specified by the user. A random number R_m is generated,. If the $R_m<P_m$ then the operation occurs. Also the location of the mutation is calculated using L_m=*INT [n-1] P_r+1]*, where n is the length of the chromosome, P_r is a random number ranging between [0,1].

The tuning of these parameters of GA is a complex and time-consuming task in practice. In this approach, the aim is to find the best combination of probability of crossover and mutation. To find the optimum values for *Pc* and *Pm*, a set of experiments was conducted. In these experiments the values for *Pc* and *Pm* are changed from 10 to 100% respectively. The number of generation was set to be as high as 10,000 generations. Usually, the optimum solution is found in less than the specified number of generations. Also the population size for these experiments was set to 20. Each combination is repeated using different seed values for random generator. Figure 7-4, illustrates different possible scenarios and the associated cost obtained using different seed values. The aim here is to identify the best experiment, which is defined by the one that gives statistically the best results.

Rep 1	1	2	3	4	5	6	7	8	9	10
	Mutations									
Crossover	100	90	80	70	60	50	40	30	20	10
100	33032079	32852670	32674235	34265080	29220887	29538047	28064129	28651207	24538336	26218954
90	35368587	32281179	32105848	27561259	26282282	28504058	26703507	26257145	20501621	27630170
80	30930083	29369738	29210220	28161707	28503952	27393195	25048032	25254155	14077802	24908883
70	30062787	28268685	28115147	29029983	29903298	26882535	26751372	22549550	20771621	25601283
60	31214554	27492648	27343325	31707261	28033100	28548540	25096034	22189397	23086310	26066356
50	32944345	26867146	26721221	33007774	26242303	29938897	28375748	23840772	24331213	26984893
40	28759407	32159777	31045016	33186800	26791822	29418957	28888364	28694879	29346144	29404836550
30	30059231	33297647	32765412	37739311	27434092	33419512	33524291	30478414	30110196	30170416346
20	35088269	33318155	33137191	38832835	28799541	34383416	32154244	30660897	30189029	31080704
10	36825118	33610566	33428014	39647826	29475328	30648481	31758257	31221135	31129026	31191284264

Rep 2	1	2	3	4	5	6	7	8	9	10
	Mutations									
Crossover	100	90	80	70	60	50	40	30	20	10
100	32537677	32442914	32806068	33585343	28929243	28949104	27561677	28103718	24167216	26375814
90	35711815	32601123	32205299	27018524	25893174	28216727	26388341	25740869	20257440	27770158
80	31818011	29624865	28653018	27643336	28370314	26879601	24705277	24758572	16213483	24911834
70	30954476	28339407	27672921	28499192	29653698	26454180	26408770	22113733	20409826	25820718
60	31186661	27095573	27020966	31095069	27701543	28103569	24737623	21765190	22682912	26097933
50	32935580	26670453	26317149	32388429	25915153	29679511	27862358	23368654	23938005	27135102
40	28906516	32143360	31000510	32554947	26269950	29057200	28356820	28141166	28924327	29532131002
30	30373123	32648382	32669505	37019929	27137127	32783462	33244589	29890884	29732727	30251842300
20	35586093	33572735	32529771	38112602	28249109	33842067	31840878	30072243	29746958	31179268
10	37003195	33818311	33142147	38868128	29003718	30204291	31421466	30620012	30709412	31266270582

Rep 3	1	2	3	4	5	6	7	8	9	10
	Mutations									
Crossover	100	90	80	70	60	50	40	30	20	10
100	32074508	30931677	30974169	32586106	27616844	28089710	26221947	27228922	22724504	2556184
90	34249908	30042958	30135069	26237847	24460145	27141673	24888108	24969344	19064831	2845667
80	30285512	27600405	27837958	26794046	26606452	26104348	23525726	23995969	16760779	2395724
70	28621938	26696802	26535674	27615487	28132238	25915793	25103275	21427771	21174155	2538737
60	30058969	25778467	25686250	30168550	26101900	27217529	23476164	21094747	21267010	2543791
50	32286078	25043084	25495151	31400295	24433842	28823286	26735391	22665561	22572712	2759166
40	28317383	30475652	29211603	31591951	25195465	27961771	26981967	27285726	27250459	2927073589
30	29729170	31074633	31281927	35887134	25945307	31882381	31420156	28971931	27986345	2993348942
20	33403373	31106730	31015555	36916927	27016432	33045483	29973738	29136485	27824039	3250542
10	35062840	31974553	31287456	37742041	27821390	29455209	29826927	29673678	28642577	3261043499

Figure 7-4 Replications conducted to find Optimum values for *Pc* and *Pm*

Average	1	2	3	4	5	6	7	8	9	10
	Mutations									
Crossover	100	90	80	70	60	50	40	30	20	10
100	32548088	32075753	32151491	33478843	28588992	28858953	27282584	27994615	23810019	26933691
90	35110103	31641753	31482072	26939210	25545200	27954153	25993319	25655786	19941297	20941832
80	31011202	28865003	28567065	27533029	27826906	26792381	24426345	24669565	15684021	20658624
70	29879733	27768298	27441248	28381554	29229745	26417503	26087806	22030351	20785200	24567550
60	30820061	26788896	26683514	30990293	27278848	27956546	24436607	21683111	22345411	22428279
50	32722001	26193561	26177840	32265499	25530432	29480565	27657833	23291662	23613977	26532696
40	28661102	31592930	30419043	32444566	26085746	28812643	28075717	28040590	28506977	33095023
30	30053841	32340221	32238948	36882125	26838842	32695118	32729679	29780410	29276423	29593924
20	34692578	32665873	32227506	37954121	28021694	33756988	31322953	29956541	29253342	24474702
10	36297051	33134476	32619206	38752665	28766812	30102660	31002217	30504942	30160339	29895214

Figure 7-5 Average minimum cost obtained

Figure 7-5, represents the average of the minimum costs obtained from the three replications. Based on these experiments it was found that statistically setting P_c and P_m to 80% and 20 % respectively results optimum solution. The following sections aims to determine the impact of different GA parameters in the overall GA search.

7.3.1 Influence of population

The population size of the genetic algorithm must be selected to increase its efficiency and to arrive at good solutions within a reasonable time. Table 7-4, shows the schedule cost obtained for different population sizes and their respective improvements. It is important to note that the aim here is to show the impact of population size on GA performance. For this reason, the initial values for P_c, P_m and N_{gen} are set to 80%, 20%, and 3000 respectively.

Table 7-4 Minimum Schedule Cost for different Population Sizes

Population Size	Average Minimum Schedule Cost	Deviation [%]
10	19671620.73	0 %
20	18476577.60	6%
30	16411274.64	17%
40	16264731.48	17%
50	16586633.96	16%
60	17424587.45	11%
70	17100409.20	13%
80	17472361.53	11%
90	17777801.94	10%
100	19237708.25	2%

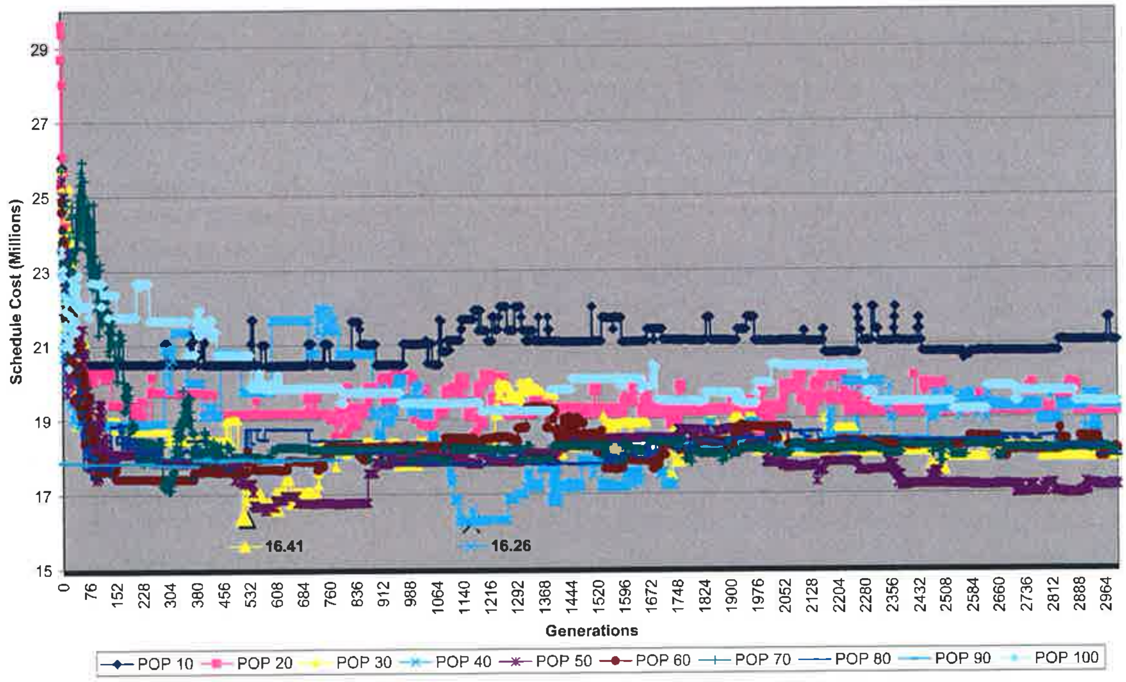


Figure 7-6 Influence of population size on GA performance

In general, it can be concluded that increasing population size improves the GA performance to find a better minimum schedule. However, higher population size does not necessary mean higher probability in obtaining the minimum cost schedule. This is mainly due to scattering of the genetic search all over the solution domain. Therefore, requiring more time to converge to an optimum solution. In addition as shown in Figure 7-6, small population size of 10 will restrict the search around small neighbourhoods resulting in poor quality solutions.

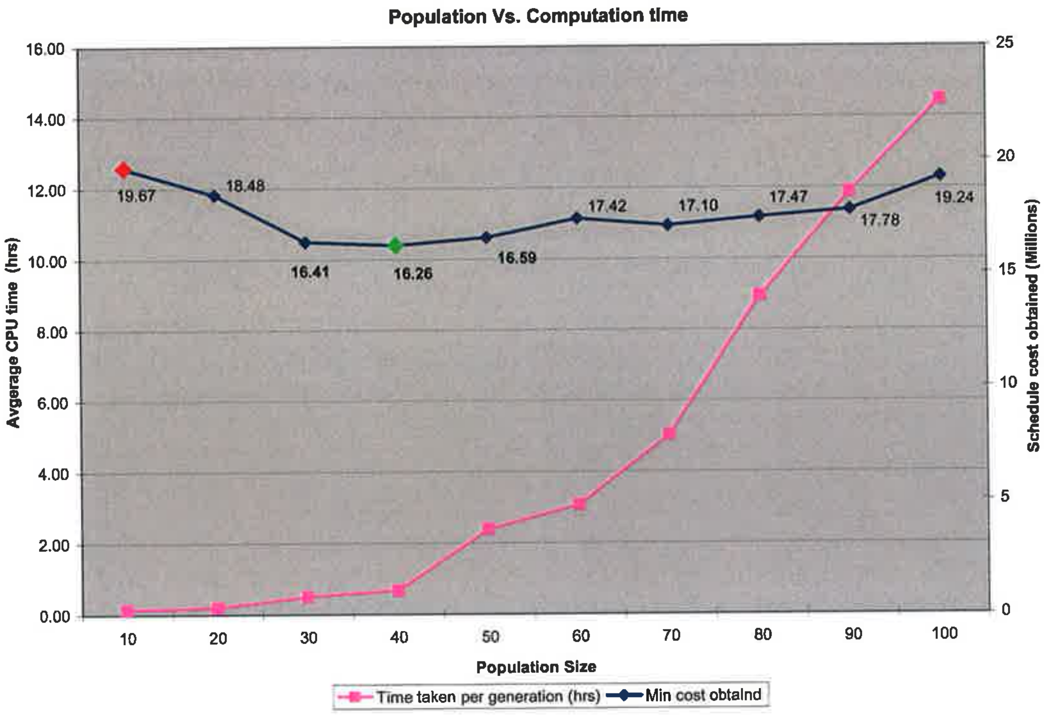


Figure 7-7 Average CPU time consumption for different population sizes

Figure 7-7, shows the impact of population size on GA performance and also it indicates the average CPU time consumption for different population sizes. The CPU time grows rapidly as the size of population increases. As shown in the above figure, the computation time picks up rapidly for population size greater than 40.

Analysing this graph suggests that using population size greater than 50 and less than 90 provide relatively good solutions but the computation time is left to be high (i.e. above 3 hrs and less than 8 hrs per generations). Large population sizes such as 90 and 100 slightly improve the solution, however the computation time is left to be as high as 12 to 14 hours per generations.

It is obvious from this graph that population size of 30, 40 and 50 provide lower cost solutions. In the case of population size 30 and 40, these require computation time less than 2 hours per generation. However, in case of population size 50, the computation time has increased sharply to above 2 hours per generations.

Finally, increasing population size from 10 chromosomes up to 30 chromosomes improves the GA performance up to 17% and in average it requires about 1 hour processing time per generations.

7.3.2 Evaluations of Selection Method Operators

The first genetic operation in the genetic algorithm is selection and a good selection scheme avoids both high selective pressure and premature convergence. In this application three selection schemes were developed based on random, roulette wheel and tournament selection methods. To investigate the impact of selection methods, an initial population of size 100 chromosomes were tested. Table 7-5, shows the minimum cost schedule obtained using each of the mentioned methods and the deviation from the best-found solution is presented. For this reason, the initial values for P_c, P_m, N_{gen} and N_{pop} are set to 80%, 20%, 2000, 20 respectively.

Table 7-5 Minimum Cost Schedule obtained from different selection schemes

Selection techniques	Average Schedule Cost	[%] Deviation
Roulette Wheel	19,671,621	0.00
Tournament	19,509,397	0.82
Random	41,292,848	-109.91

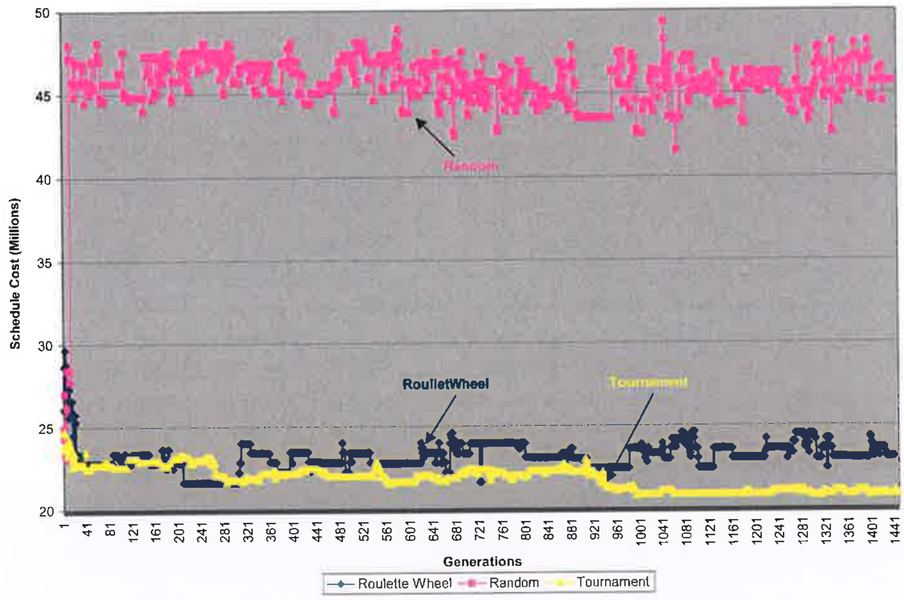


Figure 7-8 Comparison of selection methods performances

As indicated in Figure 7-8, the random search does not converge to a minimum cost schedule. In fact, as the number of generation ascends the fitness of the found schedules deteriorates. This is mainly due to the fact that fitness of the chromosomes is not considered in selection of chromosomes for genetic reproduction processes in this method. In the other hand, the roulette wheel and tournament methods converge to minimum cost chromosomes.

These methods show a higher convergence rate in the first 50 generations and thereafter the convergence rate was stayed at the same rate in case of roulette wheel method. As indicated in Figure 7-8, the tournament selection method mostly shows slightly higher and gradual convergence and it tends to find lower cost schedules than the roulette wheel method. This can be observed after 400 generations and it is more evident after 1000 generations.

In the tournament selection method, the chromosomes are selected from the selection pool for mating. An analysis was conducted to study the impact of selection pool on the GA operations. This analysis was conducted using 60 chromosomes to initiate the GA operations with. The result indicated that tournament size of 40 chromosomes leads to a better schedule cost. The performance obtained by using different size of pool and their percentage improvements are presented in Table 7-6.

Table 7-6 Impact of selection pool size on the GA Performance

Selection Pool Size	Average Schedule Cost	|%| Improvement
2	20,673,289	0
10	20,144,349	2.56
20	20,164,129	2.46
30	19,191,721	7.17
40	19,176,412	7.24
50	20,150,632	2.53
60	20,309,909	1.76

Generally, a small size of pool consists of only a few best solutions and allows limited mating. However a large selection pool contains both good and bad solutions and normally generates low quality solutions after mating. The above results indicates using tournament size 30 or 40 would be more effective in searching for the best schedule.

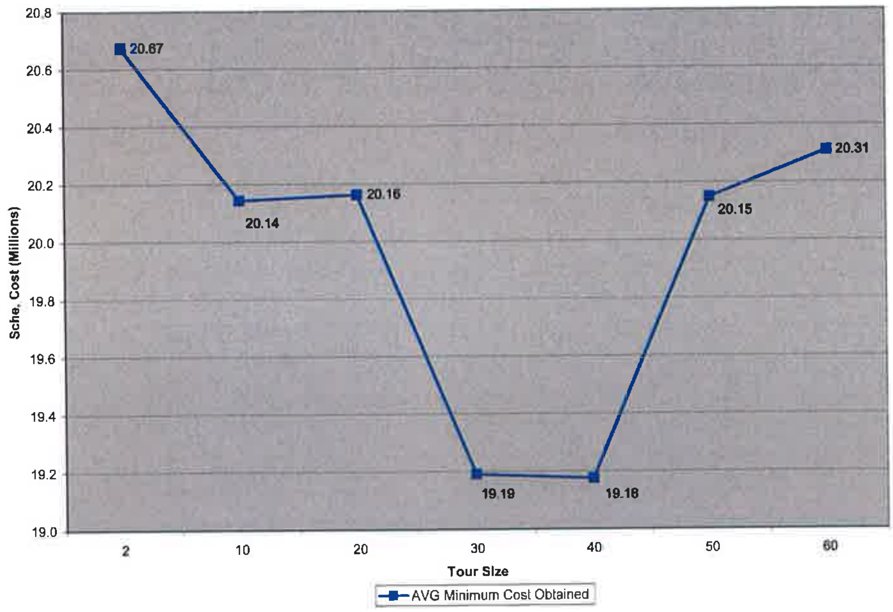


Figure 7-9 Impact of Tour size in Tournament selection

7.3.2.1 Selective Pressure

One parameter to measure the population diversity is the selective pressure. The population's diversity is characterised as small when the minimum population fitness and the average fitness are almost the same. When this condition holds, the chromosomes in the entire population are almost the same and therefore the population is converged. The high selective pressure increases the number of identical solutions and decreases the population diversity. Table 7-7, illustrates selective pressure measures for each available method. The values presented in this table are the average of three independent replications using different seed values. Each replication was run for 2000 generations. Also Figure 7-11, displays the selective pressure plot as the number of generations increase for selection schemes.

Table 7-7 Selective Pressure measures

Selection Methods	Average Selective Pressure(%)			
	Min	Max	Average	Std.
Random	49.86	75.24	63.73	(+/-) 5.25
Tournament	58.91	85.98	73.27	(+/-) 4.90
Roulette Wheel	56.04	89.73	77.48	(+/-) 4.89

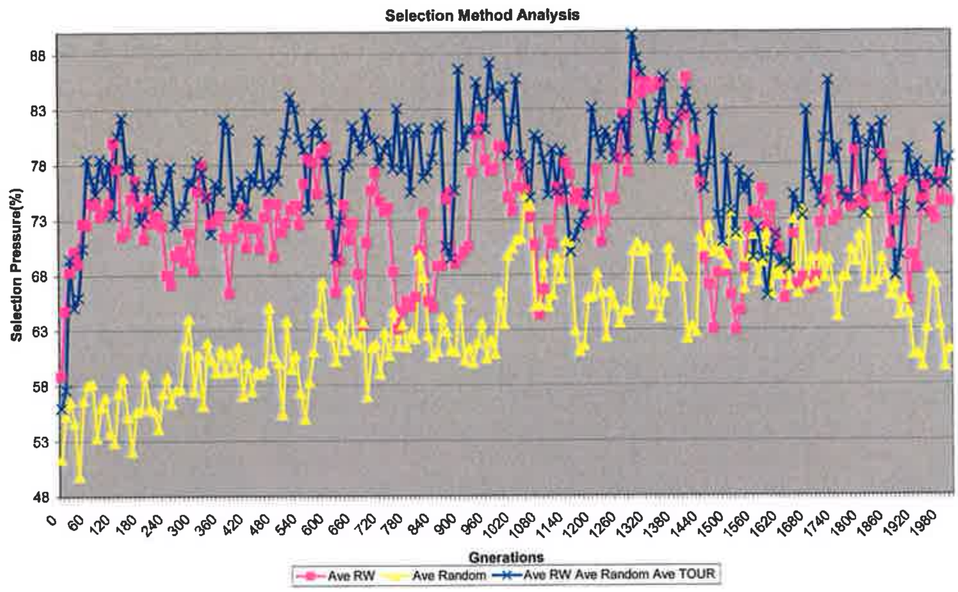


Figure 7-10 Selective pressure variations

A number of observations could be drawn from these data:

- Roulette Wheel selection method showed high selective pressure across all generations (i.e.77.48 %)
- In this method an early rise in selective pressure was evident and thereafter it was maintained throughout the genetic process.
- Tournament selection schemes, showed slightly lower selective pressure with a higher variation of selective pressure throughout the genetic process. In almost all generations the selective pressure is lower than that of the roulette wheel. There was a significant drop in selective pressure from generation 550 to 900.
- Random selection method shows the least selection pressure of all these methods. It starts with relatively lower selection pressure and it gradually increases. It reaches its peak at around generation 1050. It then felt back to lower selection pressure 1850.
- Random search provides higher diversity within the populations. In this way it introduces bad chromosomes for reproduction and it results in generation of relatively poor quality schedules.

7.3.3 Evaluation of Crossover Operators

Crossover operation is a GA reproduction process resulting in generation of new solutions and exploration of solution space. As described before there are there crossover operators developed in this application. To show the impact of these operators the initial parameters were kept constant and GA runs were set using different crossover methods. The initial parameters were set to: N_{gen}= 3000, Initial population size = 20, P_c=80% and P_m=20%. The results are indicated in Table 7-8, Table 7-9, Table 7-10 and Table 7-11. Also, Figure 7-11, Figure 7-12 and Figure 7-13 demonstrate the impact of these crossovers graphically.

- **Crossover 1, Mutation 1:**

Table 7-8 statistics for Crossover 1-Mutation 1

C1-M1	Cost
Average	15,542,588.21
Std.	[+/-] 688,570.88
Max	24,887,911.80
Min	14,077,801.94
Delta	10,810,109.86
Min @ Generation:	475

- **Crossover 2, Mutation 1:**

Table 7-9 Statistics for Crossover 2-Mutation 1

C2-M1	Cost
Average	15,679,066.09
Std.	[+/-] 451,343.98
Max	20,841,923.86
Min	14,099,143.58
Delta	6,742,780.29
Min @ Generation:	874-876

- **Crossover 3, Mutation 1:**

Table 7-10 Statistics for Crossover3-Mutation 1

C3-M1	Cost
Average	15,502,454.58
Std.	[+/-] 523,078.14
Max	22,878,944.56
Min	14,074,349.12
Delta	8,804,595.44
Min @ Generation:	937

Table 7-11 Impacts of Crossovers

GA perfomance	C1-M1	C2-M1	C3-M1
Average	15,542,588.21	15,679,066.09	15,502,454.58
std.	688,570.88	451,343.98	523,078.14
Max	24,887,911.80	20,841,923.86	22,878,944.56
Min	14,077,801.94	14,099,143.58	14,074,349.12
Delta	10,810,109.86	6,742,780.29	8,804,595.44
Min @ Generation	475	874-876	937

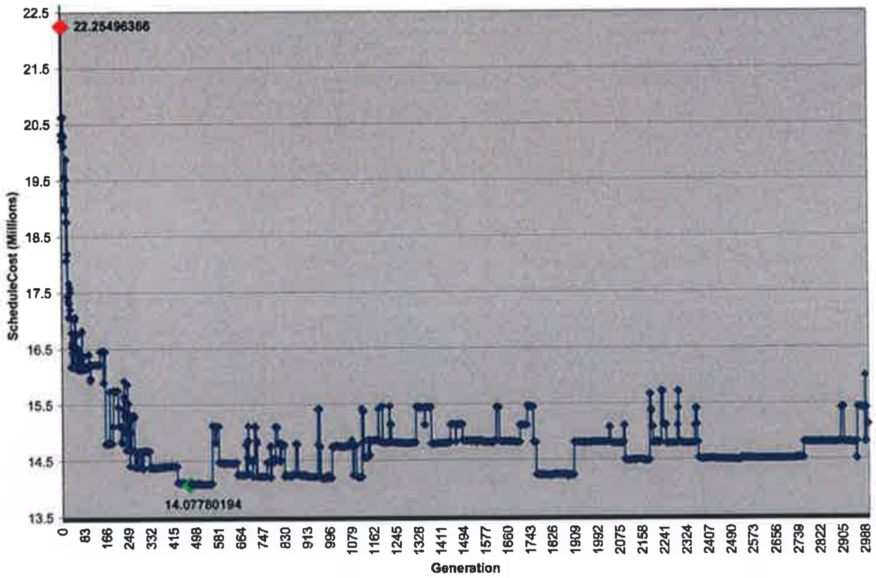


Figure 7-11 GA Performance using Crossover 1- Mutation 1

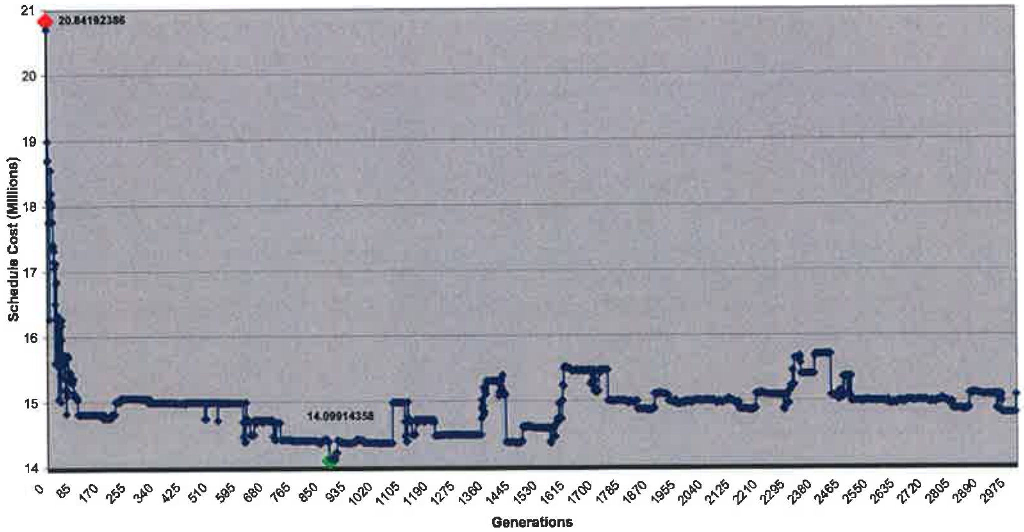


Figure 7-12 GA performance using Crossover 2, Mutation1

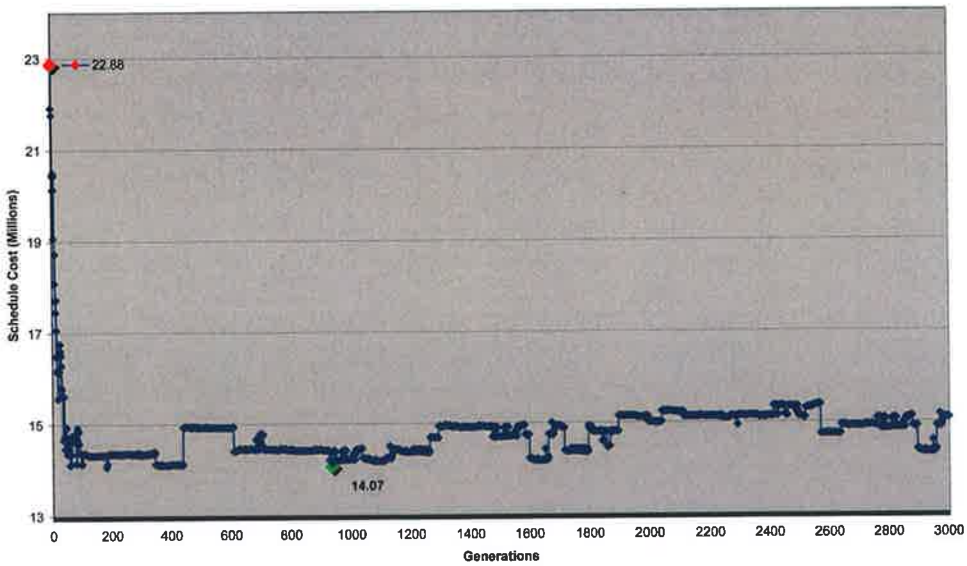


Figure 7-13 GA Performance using Crossover 3, Mutation 1

As indicated in Table 7-11, the developed crossovers converge to relatively similar schedules. This is more evident in the case of crossover type1 and type 3. However, it can be observed that crossover type 1 performs relatively better in a way that it is converged to this solution in less number of generations. Also, this method presents higher standard deviation suggesting higher diversity between generated schedules. In addition, crossover type 2 is also converged to similar schedules as found using crossover type 1 and 3. Although the schedule obtained using this method has slightly higher cost but it was reached in less number of generations.

7.3.4 Evaluation of Mutation Operators

Mutation is the other genetic operation that generates new chromosomes away from the current neighbourhood allowing the algorithm to explore a wider region. There are two mutation techniques developed in this application. In the last section different crossover methods were evaluated using mutation type 1. Similar sets of experiments were conducted using mutation type 2. The initial parameters were set as before to: Number of generations = 3000, Initial population size = 20, P_c=80% and P_m=20%. Table 7-12, Table 7-13, Table 7-14 and Table 7-15 are used to show the impact of mutation type 2 using different crossover methods. Also, Figure 7-14 Figure 7-15 and Figure 7-16 show the impact of such mutation graphically. The followings are the observations form this conduct:

- **Crossover 1, Mutation 2:**

Table 7-12 Statistics for Crossover1-Mutation 2

C2M2	Cost
Average	15,919,361.21
Std.	[+/-] 462,839.11
Max	23,657,558.88
Min	14,281,974.45
Delta	9,375,584.43
Min @ Generation:	707

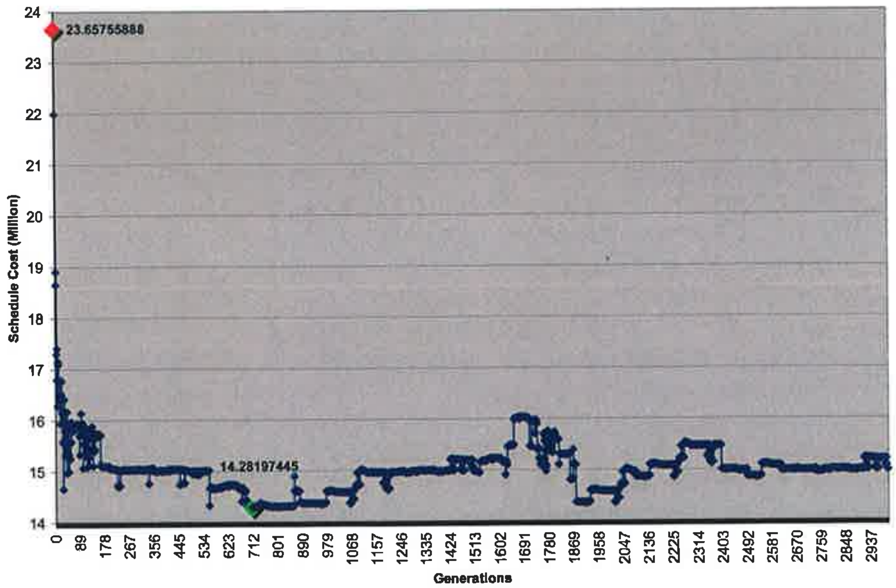


Figure 7-14 GA Performance using Crossover1, Mutation 2

- **Crossover 2, Mutation 2:**

Table 7-13 Statistics for Crossover2-Mutation 2

C2-M2	Cost
Average	19,793,508.55
Std.	[+/-] 781,675.50
Max	24,989,949.09
Min	17,598,223.04
Delta	7,391,726.05
Min @ Generation:	1009-1111

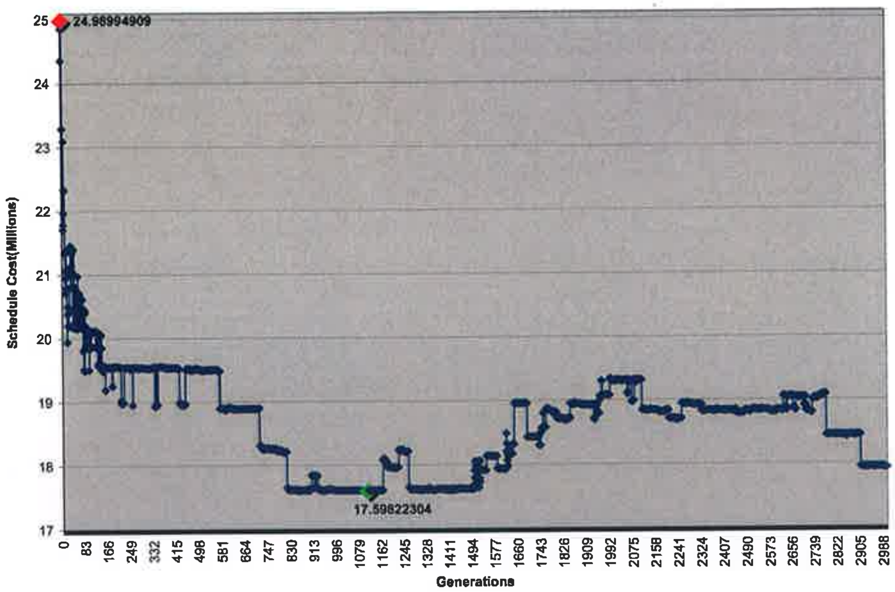


Figure 7-15 GA Performance using Crossover 2, Mutation 2

- **Crossover 3, Mutation 2:**

Table 7-14 Statistics for Crossover3-Mutation 1

C3-M1	Cost
Average	17,959,668.13
Std.	[+/-] 615,918.00
Max	22,414,896.11
Min	15,786,633.96
Delta	6,628,262.16
Min @ Generation:	477-481

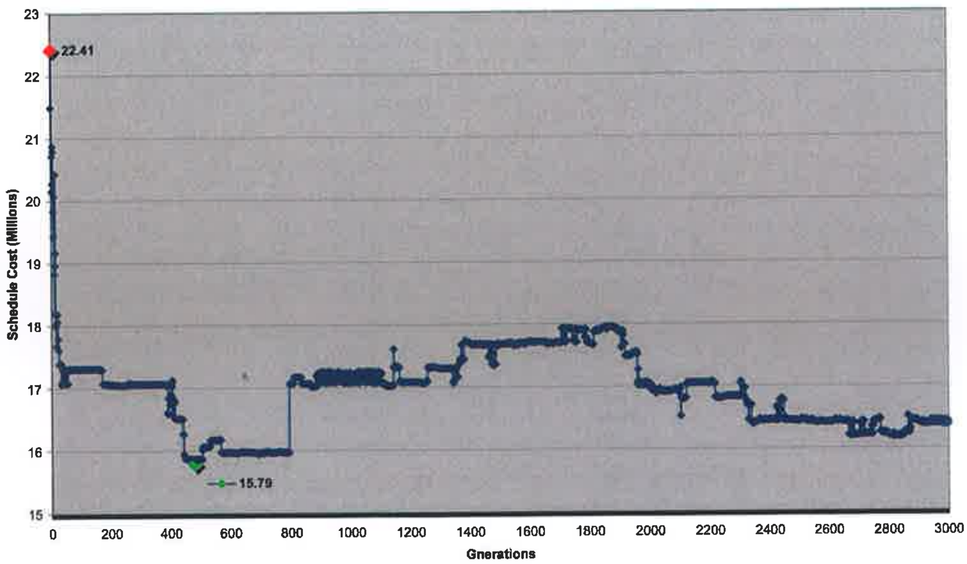


Figure 7-16 GA performance using Crossover 3, Mutation2

Table 7-15 An overall impact of crossovers and mutations on GA performance

C/M	Average	std.	Max	Min	Delta	Min @ Generation
C1-M1	15,542,588.21	(+/-) 688570.88	24,887,911.80	14,077,801.94	10,810,109.86	475
C1-M2	15,919,361.21	(+/-) 462839.11	23,657,558.88	14,281,974.45	9,375,584.43	707
C2-M1	15,679,066.09	(+/-) 451343.98	20,841,923.86	14,099,143.58	6,742,780.29	874-876
C2-M2	19,793,508.55	(+/-) 781675.50	24,989,949.09	17,598,223.04	7,391,726.05	1109-1111
C3-M1	15,502,454.58	(+/-) 523078.14	22,878,944.56	14,074,349.12	8,804,595.44	937
C3-M2	17,959,668.13	(+/-) 615918.00	22,414,896.11	15,786,633.96	6,628,262.16	477-481

Table 7-15, provides a summary of results obtained in using different crossover and mutation methods in this application. One can observe that introducing mutation type 2 allows higher diversity within the found solutions. This is more evident as the average fitness values are higher using mutation type 2. However, using this mutation did not direct the search to a better solution as the minimum cost schedules from each crossover method is higher than the solution reached using mutation type1. This indicated that mutation type 2 might require greater number of generations to converge to minimum cost schedules as gained using mutation type1.

7.3.5 Evaluation of Elitism Mechanism

Propagation of best chromosomes from current population to the next population is crucial for convergence. The number of elite chromosomes copying to the next generation is a vital factor for faster convergence but copying more chromosomes may decrease population diversity. To show the impact of elitism on GA performance, an initial population of 14 chromosomes is considered. Sequentially the number of elite chromosomes is increased to investigate this impact. Table 7-16, shows the impact of using the elitism on the best-found solution and its deviation from this schedule.

Table 7-16 the impact of Elitism on schedule cost

Number of Elite chromosomes	Average Schedule Cost	|%| Deviation
0	23,829,559	0
2	22,569,709	5
4	21,382,572	10
6	20,440,301	14
8	19,613,368	18
10	18,139,742	24
12	16,991,067	29
14	17,938,833	25

Figure 7-17, illustrates the impact of elite chromosomes on the GA search. It can be observed from this figure that as the number of elite chromosomes increases the initial convergence increases too. This initial convergence assists in finding a better minimum cost schedule. However, as the number of elite chromosomes reached the population size, the minimum cost schedule found is trapped at local minimum of 17,938,833 and thereafter the search stayed at the same value. This suggests as a result of too many elite chromosomes, the selective pressure increases and therefore it deteriorates the search and it stops at local minimum. This is further demonstrated in Figure 7-18. This scenario indicates using 12 chromosomes out of 14 provides better optimum solution.

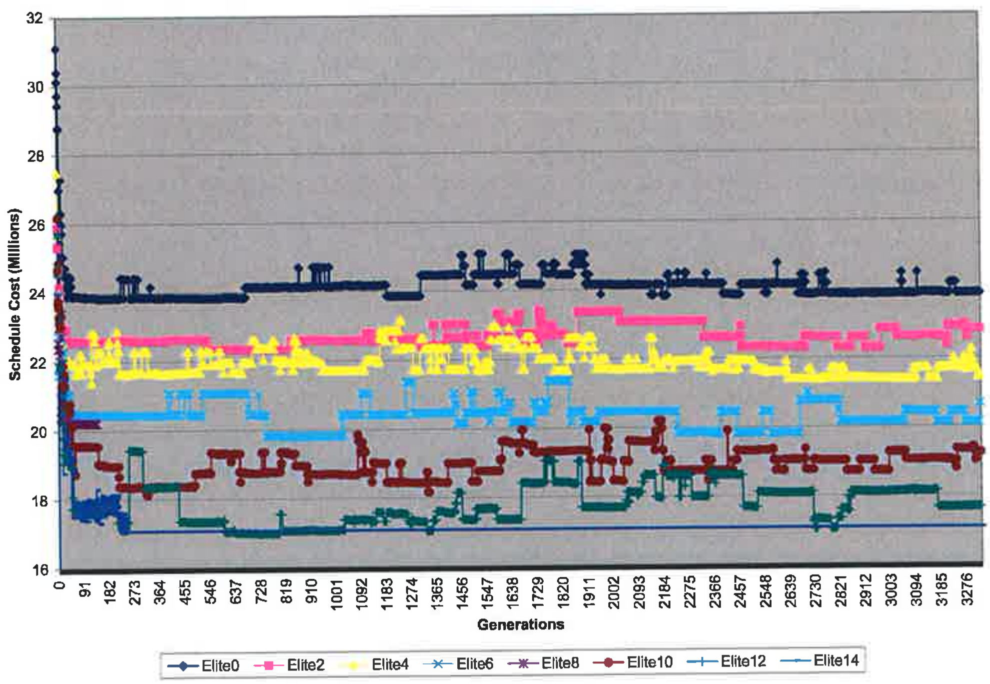


Figure 7-17 Impact of Elitism on GA performance

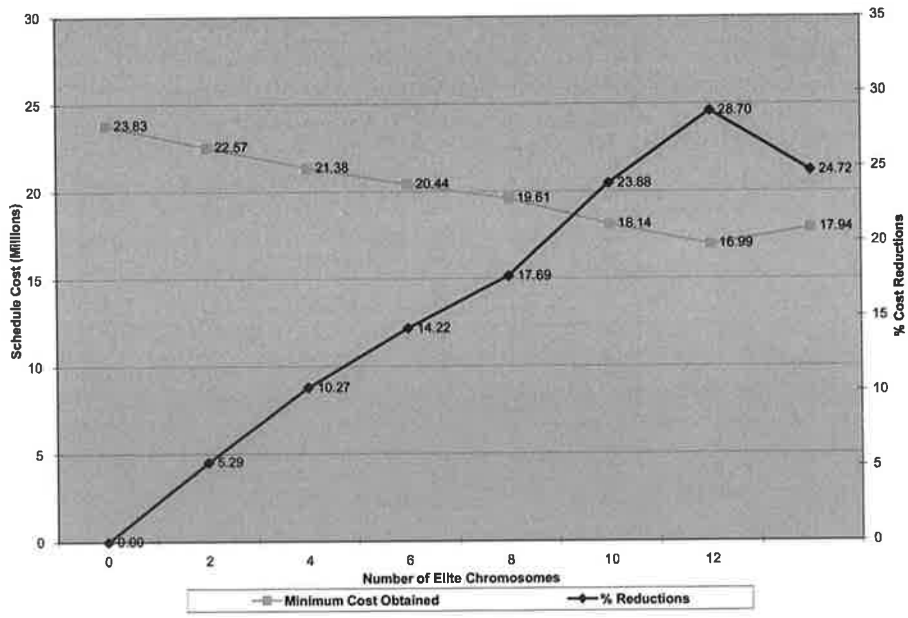


Figure 7-18 (%) schedule cost reduction using Elitism

7.4 Experimental Results and Discussions

The following points can be concluded from the conducted experiments in evaluating this GA application.

- TORA is a linear optimisation system considering supply and demand levels and the costs to transport supplies to demand. This is a single objective optimisation tool aiming to minimise the transportation cost. This approach is not concerned about resource limitations and many other real constraints that exists in the system. Using this approach the minimum cost transportation was amounted 280,959.17 Km. The obtained result from TORA could be misleading, if the difference between GA problem formulation and this application is not considered.
- The developed GA was not primarily designed based on the general transportation problem. It was designed to considered many real system constraints concerning resources and routes. Using this optimisation application the minimum transportation cost was amounted to 292,000. This value was reached after 950 generations. It is obvious the GA search is searching for the minimum transportation cost as it obtained transportation cost is close to the one from TORA, having considered the differences in problem formulations.

- Comparing the developed GA optimisation tool with random search, showed superior performance by GA tool. Using random search, the minimum schedule cost was about 21,763,302.23 unit cost. This was obtained after 10,000 generations.
- However, using GA search a minimum schedule cost was obtained amounting to 17,657,026.54 unit cost. This solution was reached after 2649 generations in less time than the random search.
- The GA outperformed the Random search and it operates closely to TORA linear optimisation tool. Considering the difference between problem formulations in GA application and TORA optimisation tool, one can conclude that the GA search tends to move to the minimum solution.
- A series of experiments were conducted to find the optimum values for P_c and P_m parameters of GA. These experiments statistically showed crossover probability of 80% and mutation probability of 20% results in minimum cost schedules. The average minimum schedule cost was about 15,684,021 unit cost.
- Also conducted experiments suggest that an increase in population size improves GA performance. However large population sizes results in scattering of the genetic search all over the solution domain and therefore deteriorating the minimum solutions and requiring further time to converge to minimum solution. In the other hand small population size restrict the search space to around the small neighbourhoods resulting in poor quality solutions.
- Further to the above point as the population size increases so does the computation time. Therefore there is a trade of between good solution and calculation time. The computation time varied from less than 1 hour per generation to 14 hrs per generations. Therefore it is vital to choose proper population size leading to minimum cost schedule in reasonable time.
- Population size 30 or 40 demonstrated good solution qualities and also they required less than 2 hours calculation times per generation.
- The selection methods as one of the main genetic operation could impact the GA performance immensely. The conducted experiments suggest the use of random selection method does not lead the search to minimum solution. This is mainly due to the fact that the fitness of the chromosomes is not considered in selection of the chromosomes for genetic reproduction process.

- The Roulette wheel and tournament selection method both converged to similar minimum cost schedules. However, it was noted that Roulette wheel method showed high selective pressure across all generations, where as tournament method showed slightly lower selective pressure allowing better exploitation of the search space.
- In tournament method, chromosomes are selected from selection pool for mating. Analysis showed an increase in the pool size would initially improve the solution quality. However large pool sizes also deteriorated the search as it allows existence of low quality chromosomes within the pool. Experiments suggested pool sizes of 30 and 40 demonstrate better quality solutions.
- When comparing crossover methods developed for this application, it was understood that these methods tend to converge to similar minimum cost schedules. However, crossover type 1 seems to reach the minimum schedules in less number of generations and this is followed by crossover type2 and then finally type 3.
- Also, when comparing mutation operators, the conducted experiments suggest that mutation type 2 introduces more diversity within the population. However, this diversity does not help to reach the minimum cost schedule in less or similar number of generations as it does in the case of mutation type 1 operator. Using type 2 operators decreases the quality of the solution obtained.
- Assessing the impact of Elitism on GA performance suggests that introducing elite chromosomes to proceeding generations improves the quality of solutions reached. However, as the number of elite chromosomes reached the population size the minimum cost schedule was trapped at local minimum. This indicates too many elite chromosomes results in high selective pressure and therefore immediate convergence to local minimum.
- The conducted experiments were used to investigate the GA performance and its general behaviour during the simulation runs. One main conclusion is the fact that this GA is performing as expected based on leanings from different GA applications. However, there are points of improvements that must be considered to further enhance the GA performance. One area that could further help in finding better quality solutions is development of new crossover and mutation mechanisms. In addition intelligent methods could be used to decrease the selective pressure during the search to prevent any premature convergence to local minimum.

7.5 Supply Chain Simulation Evaluations

Once near optimum transportation schedules are generated using the GA engine. Then the simulation model developed was used to evaluate the applicability of the optimum cost schedule. To this effect a number of tests have been conducted and the followings represent results concluded from these evaluations.

7.5.1 LPG Inventory Analysis

LPG inventory level at bottling plants plays a great role in customer service level and also on the operational cost of the supply system. For this reason monitoring the impact of the generated near optimum GA schedules on the inventory levels at bottling plants could be a great source of evaluation and analysis. For this reason, the following section illustrates the impact of transportation schedule on the inventories that are present in this supply chain. The analysis are preformed by considering LPG daily inventory levels, accumulated daily delivery patterns, demand rate, LPG volumes entering and leaving the bottling plants. The following sections cover 4 main bottling plants and the rest are presented in Appendix G.

7.5.1.1 LPG Inventory at Babol Bottling Plant

Figure 7-19, demonstrates the LPG daily Inventory level, the delivery pattern and the demand rate for the Babol Bottling plant. A number of points can be concluded as indicated in this graph:

A: This point illustrates a sharp drop in the inventory level. The inventory level started at 77 tones of LPG, then it rose to 120 and it gradually decreased to 33.64. This decline in inventory level is mainly due to the fact that there are very few deliveries from day 7 up to day 12.

B: This point indicates high number of deliveries from day 12 up to 27, resulting in high inventory level in this bottling plant. In general, if there is not enough free capacity in the inventories (i.e. at lease 18 tones), trucks are held in queues. As a result, trucks delivering to this bottling plant are faced with high queue time and possible late deliveries for latter operations.

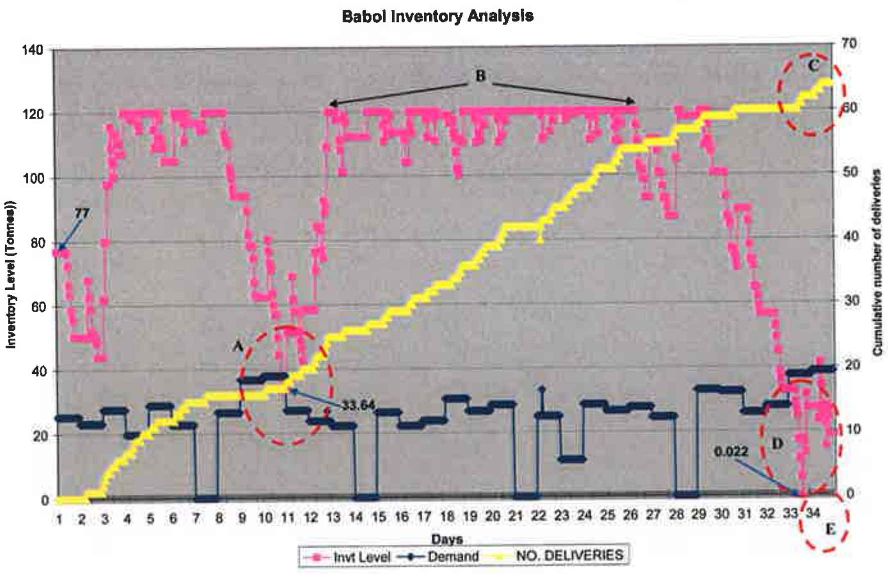


Figure 7-19 Inventory Level, Demand and Delivery analysis for Babol BP

C: Considering the schedule completion time to be 30 days and the deliveries are to be finished during this period. There are only a few deliveries from day 27 up to 30, while there are series of deliveries during day 33 and 34.

D: This point illustrates a sudden drop of inventory to a minimum point of 0.022 tones of LPG. This highlights out of stock situation where customer's demands are not met.

E: This final point suggests that the schedule completion time is above 34 days, which is above the required 30 days.

Figure 7-20, illustrates LPG volume entering and leaving the inventories in this plant. It highlights those time periods where inventory levels are critically reduced, facing possible inventory shortages and ultimately missing customer demands as described in earlier points.

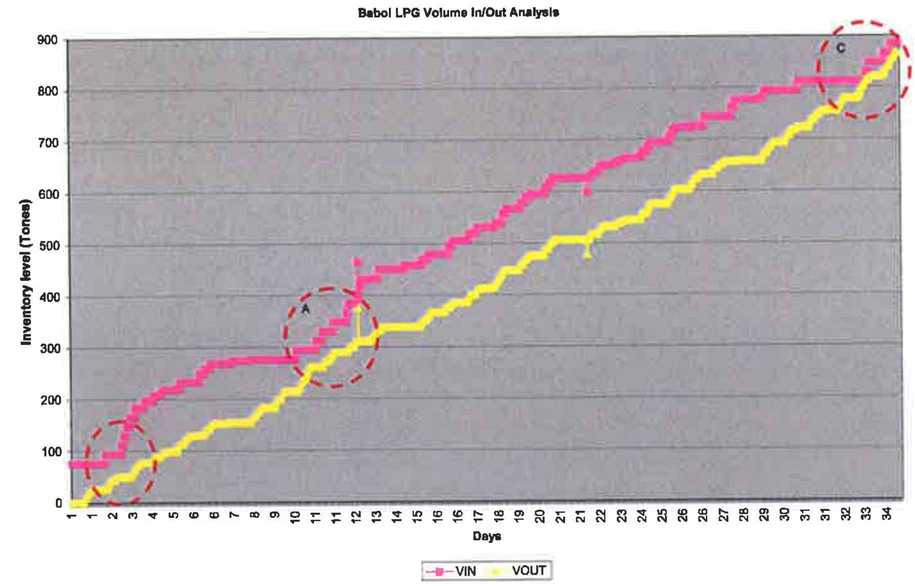


Figure 7-20 LPG volume into and out of Babol Inventories

7.5.1.2 LPG Inventory at Esfahan Bottling Plant

Similarly the inventory level at Esfahan bottling plant is investigated and looking at Figure 7-21 and Figure 7-22 the following points were observed.

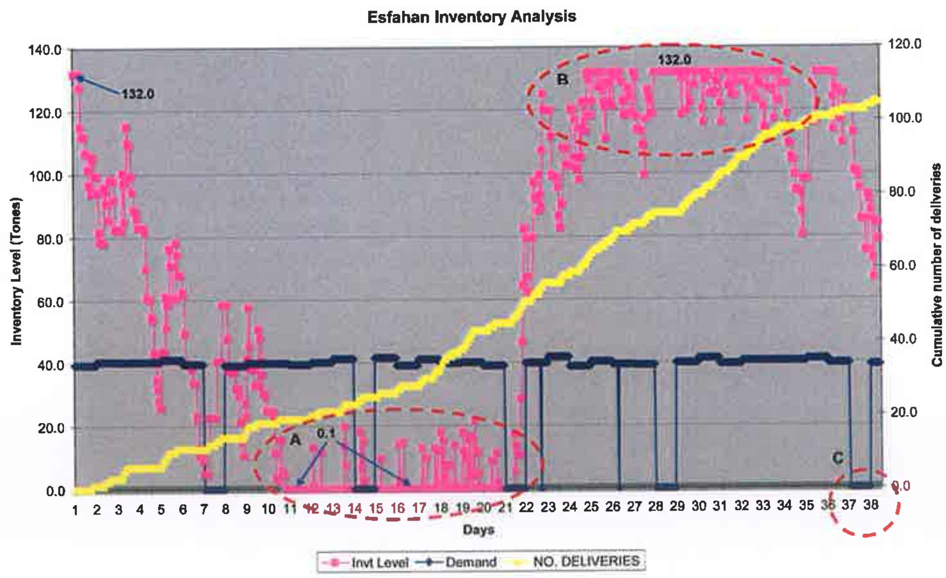


Figure 7-21 Esfahan Inventory analysis

A: This point suggests a sharp drop of Inventory level from 132 to 0.1 tones of LPG. During day 10 to 22, there are a number of deliveries pushing up the inventory level. However, the overall inventory level stays less than the demand rate and therefore resulting in LPG shortages. Figure 7-22, further illustrates the possible inventory shortage during day 10 to 21.

B: In contrary to the last point, there is a sudden increase in number of deliveries from day 22 to 34. As a result the inventory level has reached the maximum possible capacity. This would lead to further queue time and possible late deliveries for trucks.

C: This point indicates the current schedule results in late completion time above 38 days, which is much higher than the required 31 days.

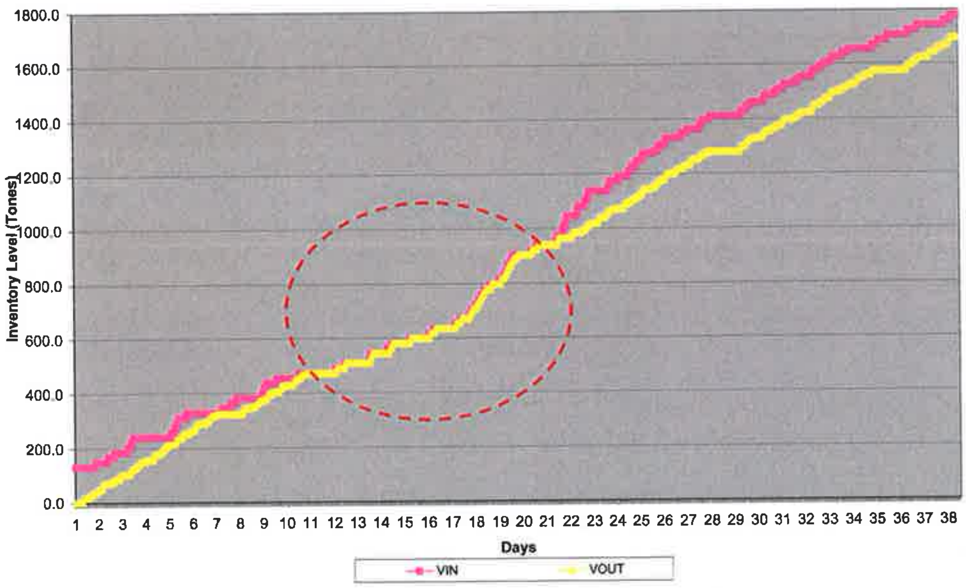


Figure 7-22 Esfahan LPG Volume analysis

7.5.1.3 LPG Inventory at Karaj Bottling Plant:

As illustrated in Figure 7-23, point A and C are used to indicate the lack of LPG supply during day 15 to 28 and from day 40 onwards. Also, schedule completion time is about 40 days as shown in point B. Also, Figure 7-24, further illustrates that the daily LPG supply is relatively the same as the LPG consumption rate. This results in low LPG availability across the delivery time period.

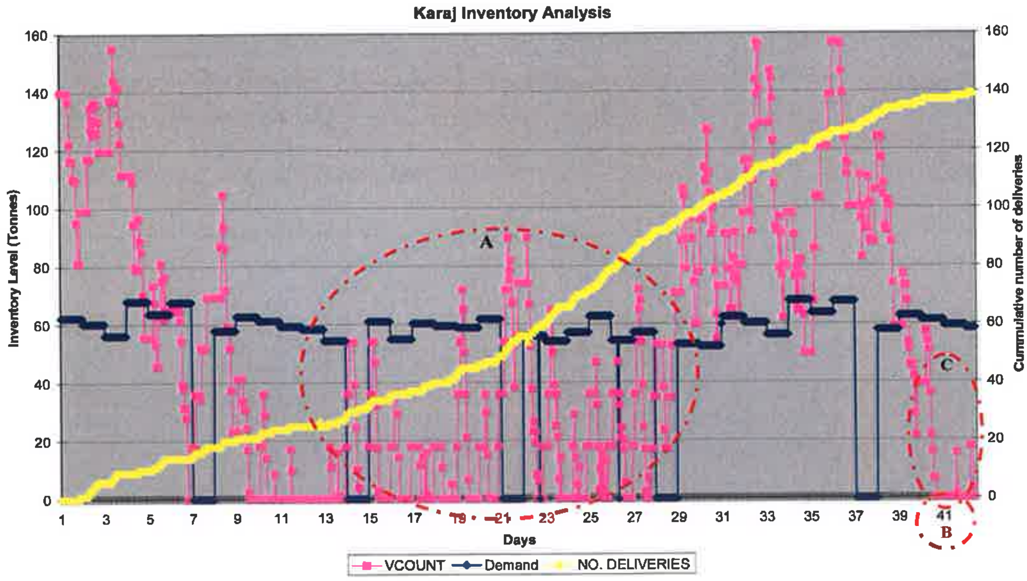


Figure 7-23 Karaj Inventory analysis

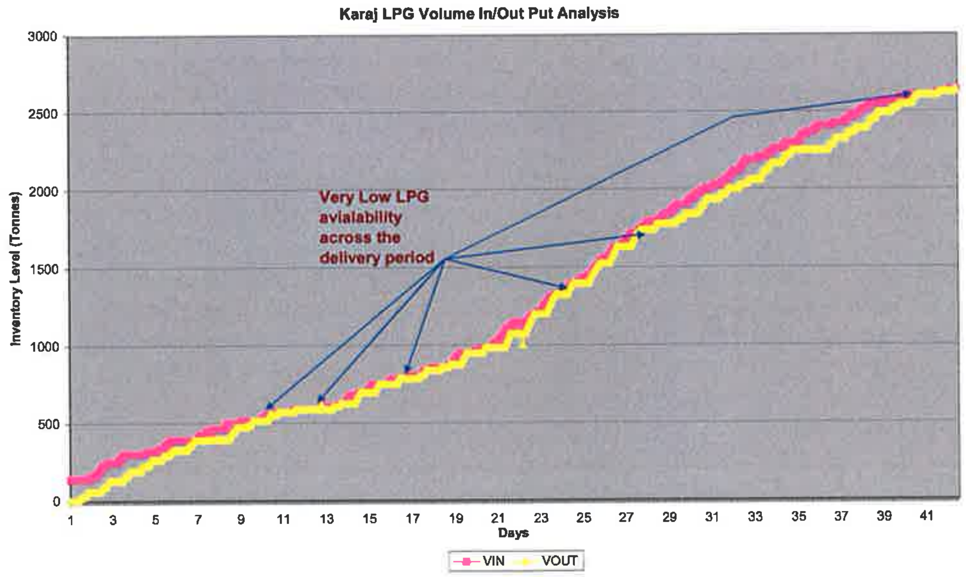


Figure 7-24 Karaj LPG volume analysis

7.5.1.4 LPG Inventory at Mashhad Bottling Plant

Figure 7-25 highlights the following points about this bottling plant.

A: There is a high number of LPG deliveries during day 4 to 21, indicating high holding inventories during this period.

B: This point suggests late deliveries made during day 33 and 34.

C: This point like the last point highlights the fact that this schedule does not finish within the required completion time of 30 days.

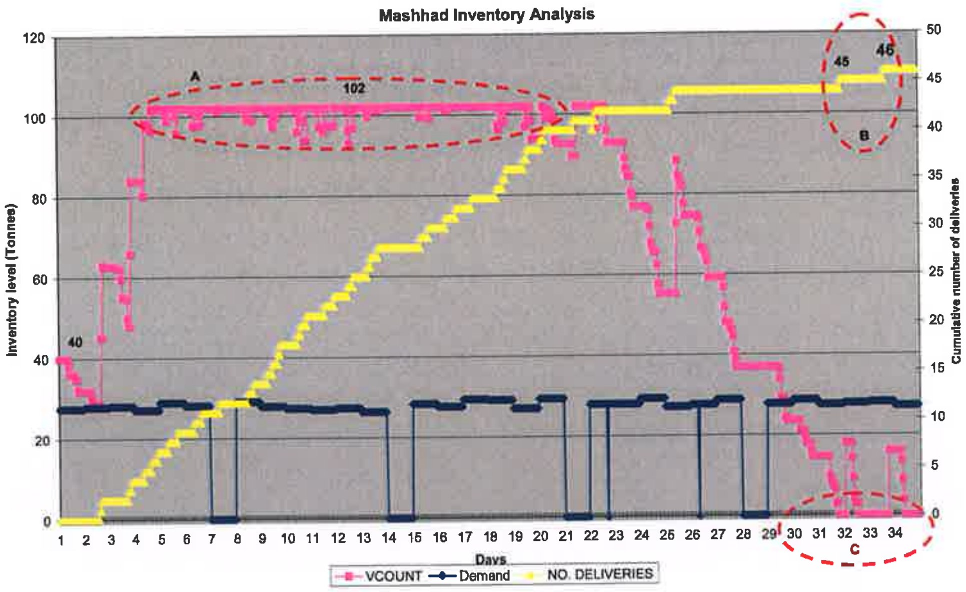


Figure 7-25 Mashhad Inventory analysis

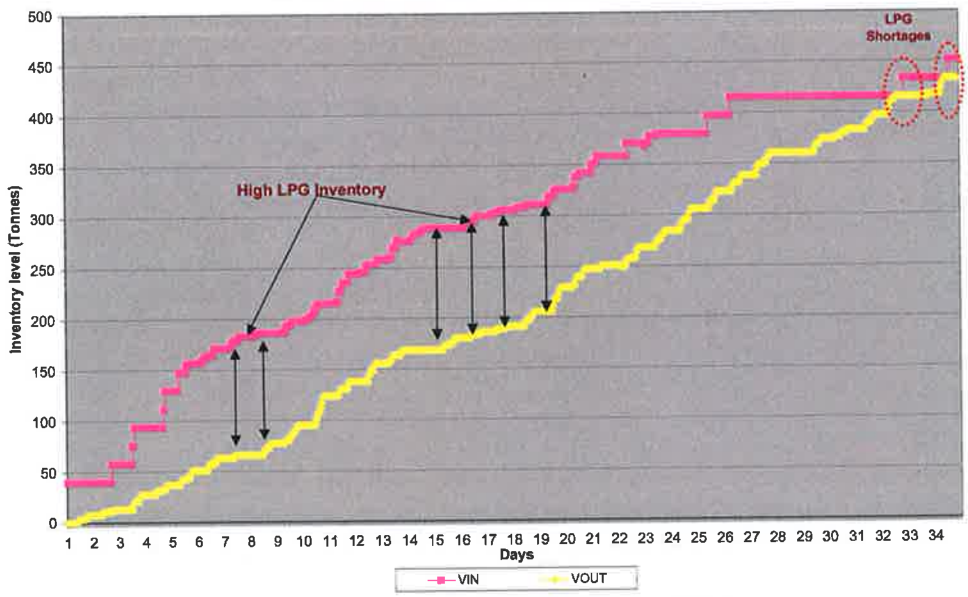


Figure 7-26 Mashhad LPG volume analysis

7.5.2 Simulation Analysis

Based on the above inventory simulation analysis, one can conclude that although the generated optimum schedule from GA presented a competitive solution, but it falls short in certain areas. These are explained as follows:

- One important point is the fact that GA presented a schedule with total completion time of almost 25 days. However, once implemented in the simulation model, there were cases showing delivery times as high as 40 days, considering the fact the time horizon for these schedules was set to 30 days.

- This prolonged deliveries, indicate the lack of stochastic and dynamic consideration in the GA optimisation tool. The behaviours such as breakdowns, repair times and queuing times that could happen in daily activities are not considered in the GA solution and therefore resulting in inaccurate specification of the final completion time for the schedule.
- In addition, although GA did not encounter any LPG Inventory shortage costs, it was evident from the above analysis that a number of Bottling plants encountered very low supply of LPG that could lead to eventual shortages and customer dissatisfactions.
- Further to the above point, the lack of LPG availabilities resulted in low service level provided by this supply system. Based on the simulation model the overall service level provided by this system was 76.8%.
- Also, there were cases indicating high numbers of deliveries are made during periods and therefore trucks delivering to the plants are faced with long queue time for unloading operation takes place.

7.5.3 Solution Improvements

To address the above discrepancies, one can conclude from the simulation model that there is a need of a constraint that limits the total deliveries per day for each bottling plant. This constraint could be based on the upper and lower limits to be specified by the user. In addition, the completion time is not sensitive enough for the current optimum schedule, therefore addition of further penalties could help the GA to search for schedule with less completion time and therefore addressing prolonged deliveries.

For this reason, constraints were added to specify lower and upper bonds on the number of LPG deliveries per day for each bottling plant exists in the system. Also further penalties were defined on the completion time. The GA was initiated again to find better solutions. The following section illustrates the impact of new modified solution for the main bottling plants considered. Appendix G provides data on the other existing bottling plant in the system.

7.5.3.1 LPG Inventory Improvement at Babol Bottling Plant

The GA constraint on the number of required deliveries for this bottling plant was updated. The lower limit on number of deliveries per day was set to 2 and the maximum was set to 4 per day. Setting these values, the GA search was set to find a better schedule. The overall results are shown in Table 7-17. Figure 7-27 and Figure 7-28 represent the impact of the new delivery schedule using the updated constraint. As shown in Figure 7-27, there are no sudden drops of LPG inventory levels leading to possible shortages. Also, the delivery pattern or LPG arrival to the bottling plant was improved resulting in continuous supply of LPG. Also, Figure 7-28 suggests that the initial inventory level or safety stock has stayed relatively at same level during this period. Finally, it is obvious that the completion time for this delivery was improved from 34 day to 30 days, which is one of the main requirements.

Table 7-17 Babol Inventory Statistics

Inventory Level LPG (Tones)	Before	After
Average	73.35	109.6
Stdev.	(+/-) 48.26	(+/-) 9.01
Max	120.00	119.8
Min	0.02	50.0

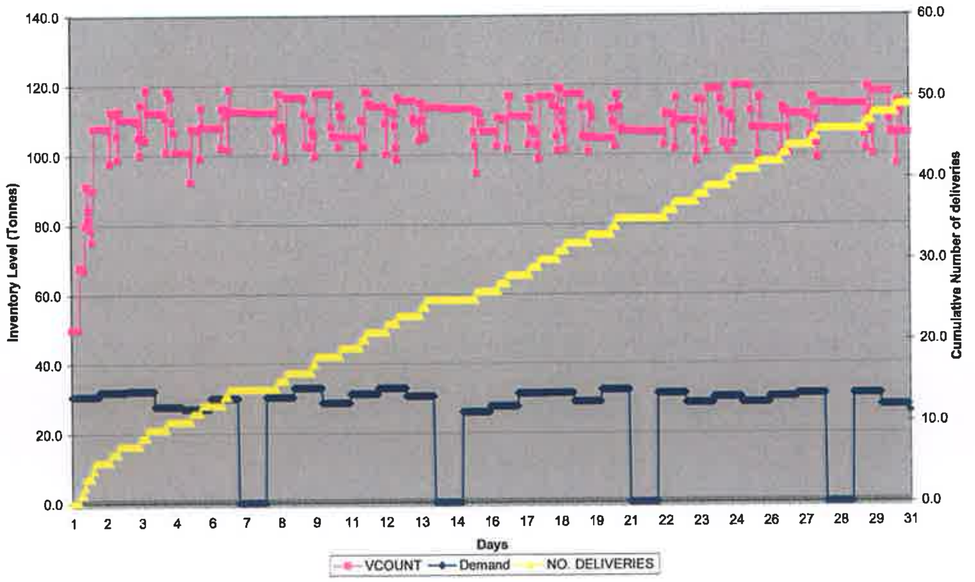


Figure 7-27 Babol Inventory analysis after Modifications

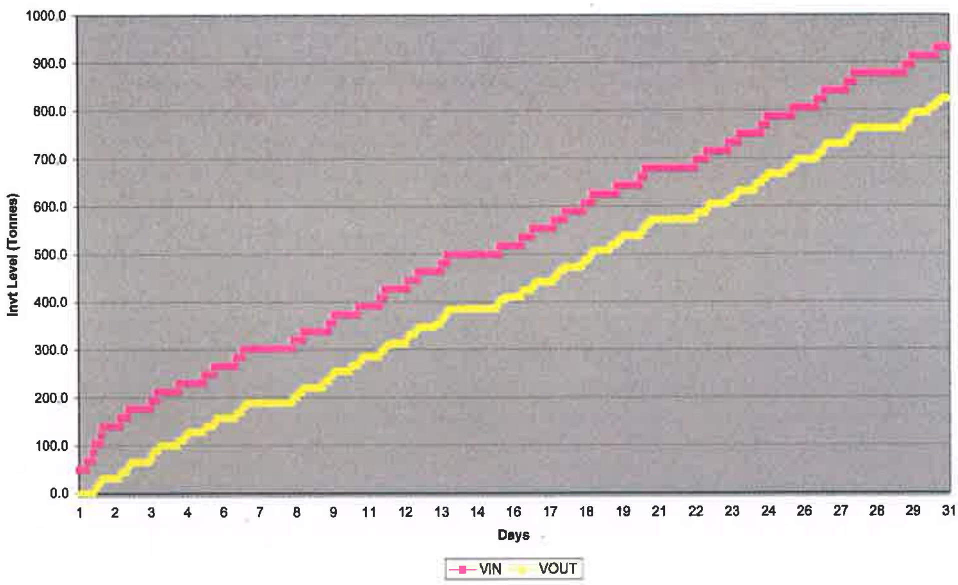


Figure 7-28 Babol LPG volume analysis after modifications

7.5.3.2 LPG Inventory Improvement at Esfahan Bottling Plant

The GA constraint on the number of required deliveries for this bottling plant was updated. The lower limit on the number of deliveries per day was set to 3 and the maximum was set to 5 per day. As shown in Figure 7-29 and Figure 7-30, the Inventory level was kept at more stable level facing no shortages during the time horizon. The improvement after this modification is further supported in Table 7-18.

Table 7-18 Esfahan inventory statistics

Inventory level LPG (Tones)	Before	After
Average	47.66	109.12
Std.	(+/-) 37.58	(+/-) 25.15
Min	132.00	52.23
Max	0.10	132.00

However, as indicated in Figure 7-29, there are no deliveries made during day 19 up until 21. This lack of delivery did not cause any out of stock situations. However, improving this gap could greatly help in reducing the total delivery cycle time.

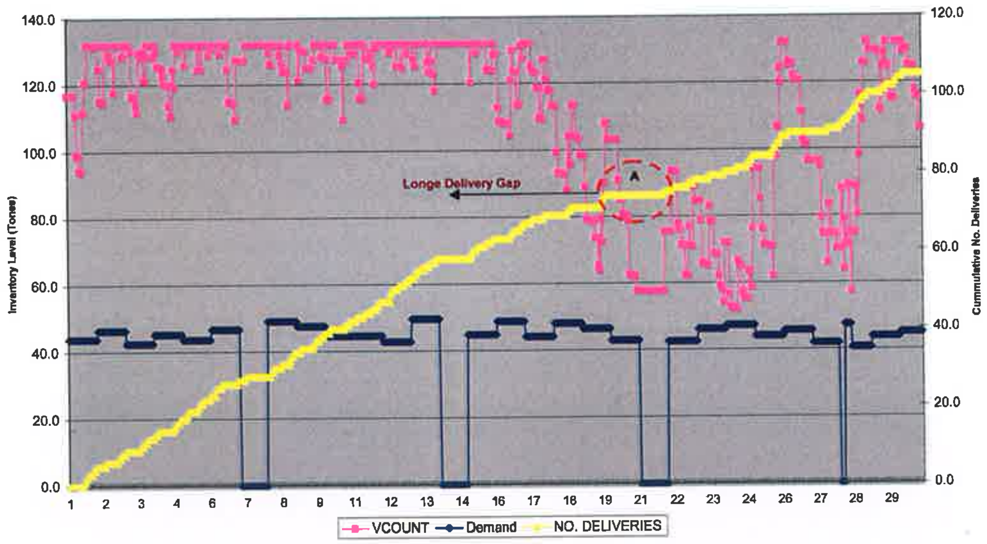


Figure 7-29 Esfahan Modified Inventory Analysis

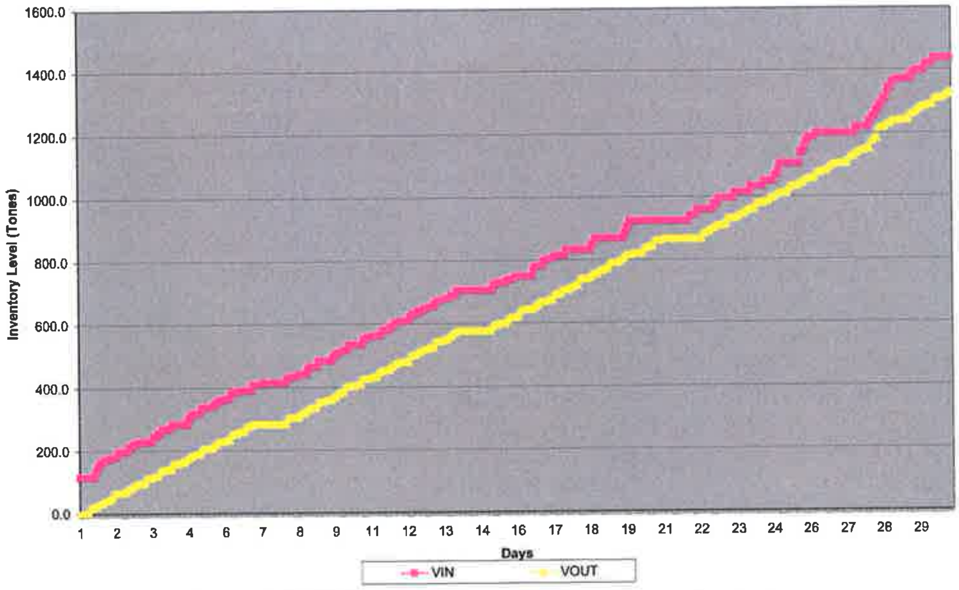


Figure 7-30 Esfahan Modified LPG Volume Analysis

7.5.3.3 LPG Inventory Improvement at Karaj Bottling Plant

The lower limit on number of deliveries per day for this plant was set to 4 and the maximum was set to 6 deliveries per day per day. Figure 7-31 and Figure 7-32 illustrate the impact of modification on the Inventory level. It is obvious that both inventory shortages and high completion times are addressed by updating these constraints. Also, Table 7-19 demonstrates these modifications quantitatively. It is obvious that the inventory level is kept high most of the time. One way to address this is to allow for lower safety stock level or to relax the daily operation limits.

Inventory Level LPG (Tones)	Before	After
Average	50.82	337.6
Stdev.	(+/-) 45.27	(+/-)47.75
Max	0.01	360.0
Min	156.86	133.5

Table 7-19 Karaj Inventory analysis

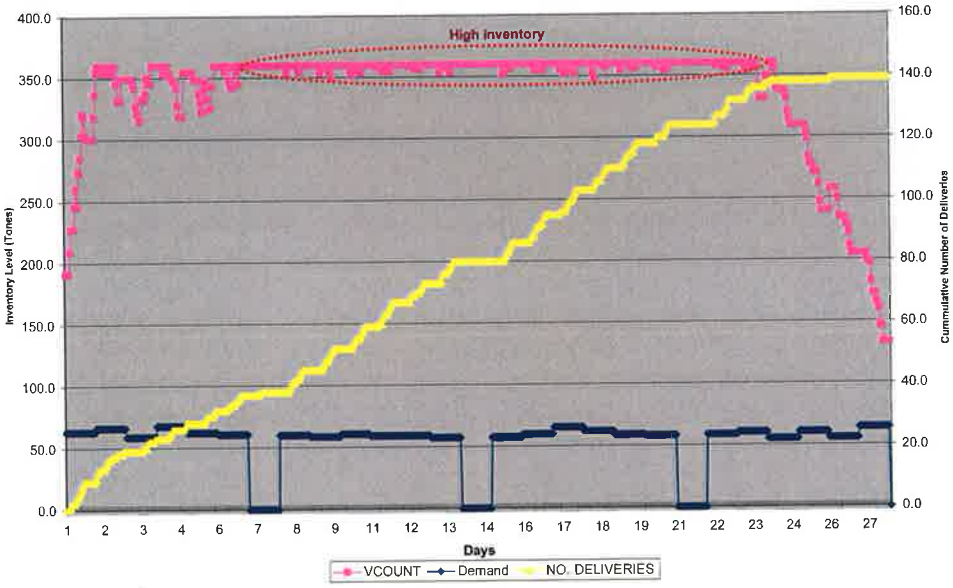


Figure 7-31 Karaj Modified Inventory Analysis

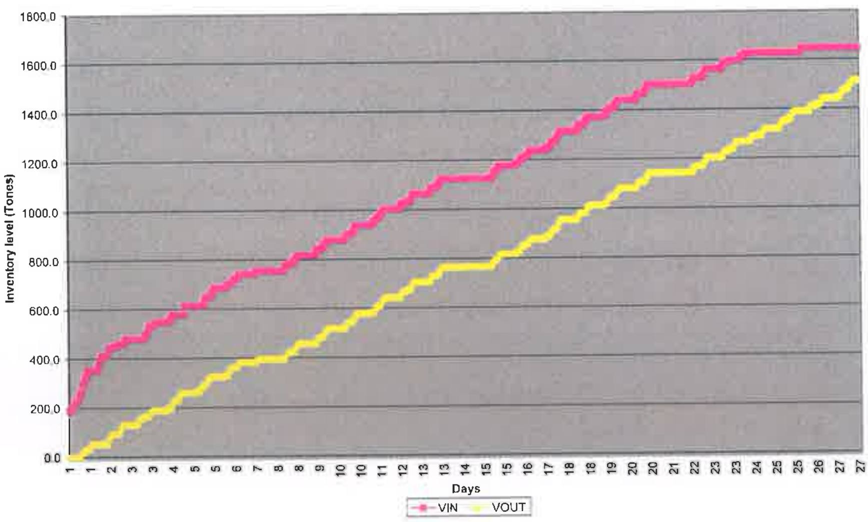


Figure 7-32 Karaj Modified LPG volume analysis

7.5.3.4 LPG Inventory Improvement at Mashhad Bottling Plant

The lower limit on number of deliveries per day for this plant was set to 1 and the maximum was set to 3 deliveries per day per day. Table 7-20 and Figure 7-33 indicate that there are no LPG shortages across the schedule. Also, all the deliveries were made within the 30 days limit. Finally, the inventory fluctuations are further normalised.

Table 7-20 Mashhad Inventory Statistics

Inventory Level LPG (Tonnes)	Before	After
Average	17.78	107.54
Std.	(+/-) 21.69	(+/-) 13.49
Max	0.00	34.22
Min	50.00	119.73

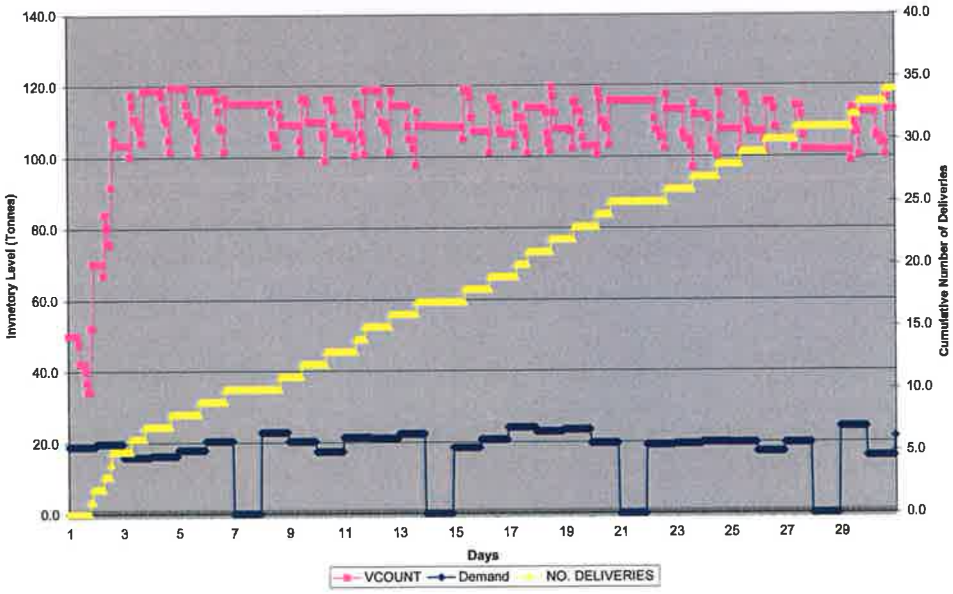


Figure 7-33 Mashhad Modified Inventory Analysis

7.6 Concluding Remarks

Obviously, using the simulator to evaluate the optimum generated schedules highlighted a number of shortfalls in implementing such plans in the real world. This approach could greatly influence the applicability of the GA generated schedules and provide better confidence to implement transportation plans across the board. Also, it is important to mention that the simulation time to run such transportation schedules was low enough as this tool was used to evaluate rather than to generate the schedules. In average each replication of such schedules would take up to 30 minutes.

As shown in the last section, setting appropriate limits on the number of daily operations and completion time constraint has made GA to search for better schedules. Therefore resulting in identification of higher quality solutions. Also, the improvement achieved at each individual bottling plant resulted in an overall improvement for the supply system modelled. The service level was improved to 95.6%. The integration of Genetic Search Engine and the Simulation Model from the supply chain system was proven to be necessary and effective. This integration allows flexibility in generating transportation schedules and it provides higher confidence in implementing such schedules in the real system.

Also, it must be mentioned that setting optimum initial parameters for the GA was fairly difficult task and time consuming. Further, enhancement of Genetic reproduction methods and application of adaptive techniques could greatly influence the GA performance.

Finally, further automation at the simulation level such as automatic feed back from the Simulator to the GA could be useful. Here, the application of Neural Networks or Expert Systems could be valuable.

Chapter 8: Conclusions

8.1 Introduction

Logistics management as a component of the supply chain includes a set of activities executed to obtain delivery of products from supply to demand locations. In general, a typical logistics manager is focussed on the effective resource planning of the distribution system so that the products are delivered on time to meet the demands. The effectiveness of this process can be evaluated based on the balance between the high service level provided by the company and the corresponding total cost in achieving the service level. Transportation is one of the key activities of logistics management and it plays an important role in both physical supply and distribution phases of the logistics. Typically transportation costs range between one-third and two-third of total logistics costs [1].

One area that determines the efficiency of transportation management is the Vehicle Routing and Scheduling activities (VRS). Managing VRS processes in supply networks is mostly perceived as uncertain, non-linear and increasingly complex. This complexity will be even more evident with increased competition and changing demands in the marketplace. Therefore, handling the VRS operations in the supply network would call for new approaches to better understand logistics processes and also new method and models to deal with logistics operations and activities in a more effective way.

The VRSP could be formulated as a NP-Hard combinatorial problem and to address such problems an extensive survey was conducted in chapter 2 and 3. Some of the early approaches used are optimisation methods such as Integer Programming (IP) and Dynamic Programming (DP) to formulate VRS problems. Also, Heuristics as alternative approaches were introduced requiring reasonable computation time in generating solutions close to the optimum. Heuristics typically use one or more strategies such as construction, improvement, component and learning to search for the optimum or near optimum solutions. Also, some widely used general-purpose heuristics to address combinatorial optimisation problems are such as tabu search; simulated annealing and genetic algorithms. Furthermore, metaheuristics as a better approach were introduced latter. The quality of solutions produced by these methods are usually much higher than that obtained by classical heuristics and algorithmic approaches.

Using the above methodologies, DSS and computer-based applications were developed to support managements with VRS problems. However, literatures [3, 71, 163] suggest that despite the large potential benefits of such systems, in practice organisation failed to adopt to these systems and very few organisations changed from their traditional practice. The majority of vehicle routing and scheduling activities are highly dependent on experienced personnel and it is performed manually especially within Small to Medium Sized companies (SMEs). A number of limiting factors reasoning the lack of computer applications in this field can be concluded from earlier survey conducted in chapter 2 and chapter 3:

- The traditional optimisation approaches such as IP and DP, have very limited applications as they only deal with problem of small size. Also, these approaches require large data storage and high computation time when faced with real world complex problems.
- The heuristic approaches were capable only to specific variation, a slight difference in the structure of the problem made the algorithm inefficient. This is mainly due to the mathematical structure of these approaches, which presents a number of disadvantages.
- Methaheuristics methods to VRSP provide solutions with higher quality than that obtained by classical heuristics. However, these methods require higher computation time and they are context dependent.

Also, further investigation on the problem formulations used to address VRS problems in logistics highlighted a number of drawbacks as specified in chapter 3:

- It is apparent that the presented heuristics, Metaheuristics and algorithmic approaches to VRS problems were not solving the problems actually faced by the vehicle fleet operators. The real problems had complexities and element of subjectivity and uncertainty measures, which could not be included in existing algorithms.
- The heuristics approaches usually address a single objective of distance minimisation. However, in the real world there may be a number of conflicting objectives based of

different factors such as cost, delivery times, fleet size, customer service, convenience factors, human factor considerations, route type and many others.

- Similarly, these problems are restricted considering vehicle capacity and distance constraints in the VRS problems, which may be only a small part of the real constraints. Constraints may include delivery time windows, the need for equitable workloads, driver shift pattern, restricted vehicle access, preferred routes, vehicle types and etc.
- In general one can observe the lack of considerations for route selections, resource utilisation, unfulfilled demands, under used capacities, reliability of deliveries, fleet size and operational cost in VRS problem formulations.
- Also, it is important to note that these approaches establish linear relationships between the considering objectives and also constraints. They fail to realise non-linearity within the objective and the constraints defined for addressing the VRS problems.
- Finally, it is important to mention that the current approaches fail to capture stochastic and dynamic nature of the logistics processes. For example uncertainty exists in the availability of raw material at supply sources and demand levels for the final items by the customers. Also resources availability which is based on it breakdown behaviours.

8.2 Contribution of the thesis

The main contribution of this thesis is the development of a hybrid decision support system to assist managers with better decision-making. In this developed approach both genetic algorithm as an optimisation method and simulation modelling have been used to meet the challenges of vehicle routing and scheduling in supply chain logistics management. The developed DSS presents a number of novel features, which have not been addressed in existing approaches. These are such as:

- The genetic search allows formulation of both single and multi-criteria VRS optimisation problems. It provides a flexible means to define linear and non-linear relations that may exist both in terms of system's objectives and also constraints. The non-linear options are developed based on the Exponential, Sin and Natural logarithm functions.

- In this application, the logistics performance measures could be also considered as part of VRSP problem formulation. This is to address the general lack of considerations for such metrics in existing VRSP approaches.

- The developed system allows incorporation of constraints from the real system. In general constraints could be modelled as either soft or hard type.

- Violation of any hard constraint results in infeasibility of the solution. In such case the solution are repaired using the repair techniques developed for this application.

- In the other hand soft constraint refers to those limits that can be violated and does not impact the feasibility of the solution. Penalty methods are used to justify the violation of the soft constraints. This allows getting some information out of infeasible solutions, and therefore extending the search space. The penalty method applies to the soft constraints here in this application.

- This application facilitates a better approach on handling route selections, resource utilisation, unfulfilled demands, underused capacities, environmental, safety, reliability of deliveries, fleet size, human factors, inventory analysis and operational cost in VRS problem which are not properly accounted for in the existing methods.

- This DSS provides a pool of solutions (i.e. near optimum schedules) allowing alternative solution considerations to assist management in decision making.

- The simulation module provides a more flexible way to incorporate stochastic, dynamic and complex relations that exists in logistics systems, which are also neglected in conventional approaches. The potential benefit in using optimisation simulation integration is the reduction of computation time and technical sophistications. The optimisation and simulation modules could be used as either a stand-alone tool or in conjunction together providing more user flexibility.

8.3 Future works

The following areas could be considered as the potential area for future work on this work.

8.3.1 Genetic algorithmsearch Enhancements

There are some potential areas that could be used to enhance the GA search performance. These areas are such as:

8.3.1.1 Selection Method:

Further research could be carried out to modify selection method to provide a better control on selective pressure. As an example methods introduced Brindle [164] such as ***deterministic sampling, remainder stochastic sampling without replacement, stochastic tournament, remainder stochastic sampling with replacement*** and ***stochastic universal sampling*** introduced by Baker [165] could be investigated for this propose.

8.3.1.2 Genetic Reproduction Methods:

The proper choice of crossover and mutation operators is critical for the successful implementation of genetic algorithms. Different crossover and mutation operators are suitable for different problems even for different stages of the GA process in a problem. Determining which crossover and mutation operators should be used and also their respective rates are quite difficult tasks and they are usually performed by trial and error. To this effect, it could be very useful to further study on the possible crossover and mutation methods to assist in better exploration and exploitation of the solution space. Also, possible investigation on using multi parents for genetic reproduction process could be useful as suggested in [166]. Also, Appendix B provides further crossover and mutation types that could be used to enhance the GA performance.

Another interesting area would be investigating principle of ***Dynamic Genetic Algorithm (DGA)*** [167]. This method simultaneously uses more than one crossover and mutation operators to generate the next generation. Also, the ratios of these operators are changed along with the evaluation results of the respective offspring in the next generation.

8.3.1.3 Adaptive Techniques:

Also investigation on adaptive techniques for GA control parameters could greatly improve GA performance. The adaptive techniques could be used to control the selective pressure to prevent premature convergence. Appendix B provides some works in this filed which are worth further investigation to improve this application.

8.3.1.4 Multiple Population

The use of multi population instead of current single population could bring some improvements for this GA application. Researchers [168, 169] have found ***Multi Population GA*** (MPGA) to be more effective both in speed and solution quality. Some features of this approach are:

- They would shorten the number of generations needed to find the optimal or near optimal solution.
- They usually can find more than one optimal solution,
- They have more resistance to premature convergence.

Therefore, further investigation on this method could contribute to the GA performance. Appendix B provides further details on this approach.

Several researchers have shown parallel/multi-population GA for different applications yield better performance than single population implementations [170, 171]. To this effect, further study on how to formulate the objective function to use the concept of Parallel GA would help reduce the computation time in this work.

8.3.1.5 Pareto- Genetic Search developments

Some initial work has been conducted to use the concept of Pareto- optimality principle in this application. However, it is not totally completed. One of the advantages of this system would be the GA becomes independent from cost calculation parameters. This could provide better accuracy in problem formulation. One potential area of further research would be to complete this algorithm and to perform further comparison with current developed approach.

8.3.1.6 Independent GA

Currently, the GA search engine communicates intensively with SQL server to transfer and store data. This application could be further improved to function independently from SQL server and therefore improving computation time.

8.3.1.7 Object library

Using object oriented concepts; one could develop an object library to further provide flexibility in defining any type of constraints and objective functions for this application. This requires great programming knowledge.

8.3.2 Expert system

Currently, a human expert uses the data from the simulation analysis to update constraints and objective function cost parameters for re-optimisation process. This operation could be further improved using Expert systems or Neural Networks to learn from the simulation model and adapt the changes in the GA search module.

References

1. Ballou, R.H., ***Bussiness logistics management.*** Prentice Hall, (1992), Englewood Cliffs, N.J, ISBN: 0131055453.
2. The Economist., **A moving story, special report on logistics**, The Economist. (December 7th 2002).
3. Waters, C.D.J., **Expert systems for vehicle scheduling.** Journal of Operational Research Society, Vol. 41, Issue 6, (1990), PP: 505-15.
4. Karp, R.M., **Reducibility among combinatorial problems.** in Miller, R.E. and Thatcher, J.W. (Eds.), Complexity of computer computations, (1972), PP:85-103.
5. Psarras, J., Stefanitsis, E., Cristodoulou, N., **Theory and methodology, combination of local search and CLP in the vehicle -fleet scheduling problem.** European Journal of Operational Research, Vol. 98, (1997), PP: 512-21.
6. Bodin, L., Golden, B., Assad, A., Ball, M., **Routing and scheduling of vehicles and crews, the state of the art.** An International Journal of Computers and Operations Research, Vol. 10, Issue 2, (1983), PP: 63-211.
7. Eilon, S., Watson-Gandy, C.D.T., Christofides, N., **Distribution management: Mathematical modelling and practical analysis.** The Statistician, Vol. 21, Issue 1, (1972), PP:89-90, ISBN: 10.2307/2986859.
8. Chen Wen, Eberhart, R.C., **Genetic algorithm for logistics scheduling problem.** Proceedings of the 2002 Congress on Evolutionary Computation, Vol. 1, Issue 1, (2002), PP: 12-6, ISBN: 0 7803 7282 4.
9. Papadimitriou, C.H., Steiglitz, K., ***Combinatorial optimisation: Algorithms and complexity.*** Prentice Hall, (1982), Englewood Cliffs, N.J., USA, ISBN: 0 13 152462 3.
10. Hoffman, K., **Combinatorial and integer optimisation.** George Mason University, http://citeseer.ist.psu.edu/247224.html.
11. Cohon, J.L., ***Multi-objective programming and planning.*** Academic Press, (1978), New York, ISBN: 0121783502.
12. Sait, S.M., Habib, Y., **Iterative computer algorithms with applications in engineering, solving combinatorial optimisation problems.** IEEE Computer Society, (1999), ISBN: 0-7695-0100-1.

13. Dantzig, G.B., Fulkeson, D.R., ***Minimising the number of tankers to meet a fixed schedule.*** Naval Res Logistics Quart, Vol. 1, (1954), PP: 217-22.

14. Puterman, M.L., ***Markove decision processes: Discrete stochastic dynamic programming.*** Wiley & Sons, (1994), New York, ISBN: 0471619779.

15. Colin, R.R., ***ADVANCED TOPICS IN COMPUTER SCIENCE: Modern heuristic techniques for combinatorial problems.*** McGraw-Hill International (UK) Limited., (1995), UK, ISBN: 0 7 709239 2.

16. Foulds, L., ***Combinatorial optimisation for undergraduates.*** Springer-Verlag, (1984), New York, ISBN: 038790977X.

17. Metropolis, N., Rosenbluth, A.W., Teller, A.H., Teller, E., **Equation of state calculation by fast computing machines.** Journal of Chemistry and Physics, Vol. 21, (1953), PP:1087-91.

18. Nearchou, A.C., **Flow-shop sequencing using hybrid simulated annealing.** Journal of Intelligent Manufacturing, Vol. 15, (2004), PP: 317-28.

19. Francois, C.J., Gendreau, M., Laporte, G., **A tabu search heuristic for periodic and multi-depot vehicle routing problems.** Networks, Vol. 30, Issue 2, (Sept. 1997), PP: 105-19, ISBN: 0028-3045.

20. Glover F, Laguna M. **Tabu search. Modern heuristic techniques for combinatorial problems**. In: C.R. Reeves Ed. New York: Blackwell Scientific Publications; 1993. p. 70-150.

21. Alter, S., ***Information systems: A managment prespective.*** 2nd ed. Benjamin/Cummings Company, (1996), Menlo Park, CA, ISBN: 0805324305.

22. Magnanti, T.L., **Combinatorial optimisation and vehicle fleet planning: Perspectives and prospects.** Networks, Vol. 11, (1981), PP: 179-213.

23. Gayialis, S.P., Tatsiopoulos, I.P., **Design of an IT-driven decision support system for vehicle routing and scheduling.** European Journal of Operational Research, (2003).

24. Mathirajan, M., Meenakshi, B., **Experimental analysis of some variants of VOGEL's approximation method.** Asia-Pacific Journal of Operational Research, Vol. 21, Issue 4, (2004), PP: 447-62.

25. Dantzig, G.B., Ramser, J.H., **The truck dispatching problem.** Mgmt Sci, Vol. 6, (1959), PP:80-91.

26. Marien, E.J., **The application of linear programming to distribution system oriented towards service.** International Journal of Physical Distribution. Mat. Mgmt., Vol. 3, (1973), PP: 191-204.

27. Baghchi, P.K., Nag, B.N., **Dynamic vehicle scheduling: An expert system approach.** International Journal of Physical Distribution and Logistics Management, Vol. 21, Issue 2, (1991), PP: 10-8.

28. Gendreau, M., Laporte, G., Semet, F., **A tabu search heuristic for the undirected selective travelling salesman problem.** European Journal of Operational Research, Vol. 106, (1998), PP: 539-45.

29. Lin, S., **Computer solutions to the travelling salesman problem.** Bel System Technical Journal, (1965), PP: 2245-69.

30. Christofides, N., Eilon, S., **An algorithm for the vehicle dispatching problem.** Operations Research Quart, Vol. 20, (1969), PP: 309-18.

31. Lin, S., Kernigham, B.W., **An effective heuristic algorithm for the travelling salesman problem.** Operation Research, Vol. 21, (1973), PP: 498-516.

32. Christofides, N., **Modern methods of vehicle scheduling.** Freight Managment, (November 1979), PP: 53-6.

33. Russell, R.A., **An efficient heuristic for the M-tour travelling salesman problem with some side conditions.** Operation Research, Vol. 25, (1977), PP: 517-24.

34. Mole, R.H., Jameson, S.R., **A sequential route building algorithm employing a general saving criterion.** Operation Research Quarterly, Vol. 27, (1976), PP: 503-11.

35. Watson Grandy, C.D.T., **Advances in distribution, A paper given at OR society conference.** York University, (October 1978).

36. Gillett, B., Miller, L., **A heuristic algorithm for the vehicle dispatch problem.** Operation Research, Vol. 22, (1974), PP: 340-9.

37. Golden, B., Assad, A., Levy, L., Gheysens, F., **The fleet size and mix vehicle routing problem.** Management Science and Statistics Working Paper No. 80-020, University of Maryland at College Park, (1982).

38. Clark, G., Wright, J.W., **Scheduling of vehicles from central depot to a number of delivey points.** International Journal of Operations Research, Vol. 12,(1964), PP:568-81.

39. Bodin, L., Rosenfield, D., Kydes, A., **UCOST.A micro approach to a transit planning problem.** Journal of Urban Analysis, Vol. 5, Issue 1, (1978), PP: 47-69.

40. Smith, B., Wren,A. ,**VAMPIRES and TASC: Two successfully applied bus scheduling programs.** Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew scheduling, (1981), PP: 97-124.

41. Wren, A., Holiday, A. , **Computer scheduling of vehicles from one or more depots to a number of delivery points.** Operation Research Quarterly, Vol. 23, (1972), PP: 333-44.

42. Rochat, Y., Sement, F., **A tabu search for delivering pet food and flour in Switzerland.** Journal of the Operations Research Society, Vol. 45, Issue 11, (1994), PP: 1233-46.

43. Solomon, M.M., **Algorithms for the vehicle routing and scheduling problems with time window constraints.** Operation Research Society of America, Vol. 35, Issue 2, (1987).

44. Laporte, G., Gendreau, M., Potvin, J.Y., Semet, F., **Classical and modern heuristics for the vehicle routing problem.** International Transactions in Operational Research, Vol. 7, (2000), PP: 285-300.

45. Toth, P., Vigo, D., **Granular tabu search.** Working Paper DEIS, University of Bologna, (1998).

46. Potvin, J.Y., Rousseau, J.M., **Theory and methodology: A parallel route building algorithm for vehicle and scheduling problem with time windows.** European Journal of Operational Research, Vol. 66, (1993), PP: 331-40.

47. Duhamel, C., Potvin, J.Y., **A tabu search heuristic for the vehicle routing problem with backhauls and time windows.** Journal of Transportation Science, Vol. 31, Issue 1, (February 1997).

48. Russell, R.A., **Hybrid heuristics for the vehicle routing and scheduling problem with time windows.** Transportation Science, Vol. 29, (1995), PP: 156-66.

49. Thongiah, S.R., Potvin, J.Y., Sun, T., **Heuristic approached to vehicle routing with back hauls and time windows.** Computers and Operation Research, Vol. 23, Issue 11, (1996), PP: 1043-57.

50. Vukadinovic, K., Teodorvic, D., Pavkovic, G., **Theory and methodology, a neural network approach to the vessel dispatching problem.** European Journal of Operational Research, Vol. 102, (1997), PP: 473-87.

51. Krolak, P., Felts,W., Marble,G., **A man-machine approach toward solving the travelling salesman problem.** Commun ACM, Vol. 14, (1971), PP: 327-34.

52. Krolak, P., Felts,W., Nelson,J., **A man-machine approach toward solving the generalised truck dispatching problem.** Transportation Sci., Vol. 6,(1972), PP:149-70.

53. Sik Jo, G., Jung, J.J., Yang, C.Y., **Expert system for scheduling in an airline gate allocation.** Expert Systems with Applications, Vol. 13, Issue 4, (1997), PP: 275-82.

54. Hopfield, J.J., Tank, D.W., **Neural computation of decision in optimisation problems.** Biological Cybernetics, Vol. 25, (1985), PP: 141-152.

55. Ghaziri, H., **Supervision in the self-organising feature map: Application to the vehicle routing problem in Metaheuristics: Theory and applications.** I.H. Osman and J.P. Kelly, (1996), PP: 651-60.

56. Wang, J., **A recurrent neural network for solving the shortest path problem.** Proceedings of IEEE International Symposium Circuits and Systems, Vol. 6,(1994), PP:319-22.

57. Smith, K.A., **Neural networks for combinatorial optimisation: A review of more than a decade of research.** INFORMS Journal on Computing, Vol. 11, Issue 1, (1999).

58. Potvin, J.Y., Shen,Y., Rousseau, J.M., **Neural network for automated vehicle dispatching.** Computer and Operations Research, Vol. 19, Issue 3-4, (1992), PP: 267-76, ISBN: 0305-0548.

59. Teodorovic, D., **Fuzzy logic systems for transportation engineering: The state of the art.** Transportation Research Part A, Vol. 33, (1999), PP: 337-64.

60. Teodorovic, D., Kikuchi, S., **Application of fuzzy sets theory to the saving based vehicle routing algorithm.** Civil Engineering Systems, Vol. 8, (1991), PP: 87-93.

61. Teodorovic, D., Pavkovic, G., **The fuzzy set theory approach to the vehicle routing problem when demand at nodes is uncertain.** Fuzzy Sets and Systems, Vol. 82, (1996), PP: 307-17.

62. Xue, D., Wang, H., Norrie, D.H., **A fuzzy mathematics based optimal delivery scheduling approach.** Computers in Industry, Vol. 45, (2001), PP: 245-59.

63. Aminu, U.F., Eglese, R.W., ***A constraint programming approach to the Chinese postman problem with time windows.*** Computer and Operations Research, Vol. 33, (2006), PP: 3423-32, www.sciencedirect.com.

64. Wang, Y.Z., **Using genetic algorithm methods to solve course scheduling problems.** Expert System Applications, Vol. 25, (2003), PP: 39-50.

65. Lee, S.M., Moore, L.J., **Optimising transportation problems with multiple objectives.** AIIE Transactions, (1973), PP: 333-8.

66. Cadenas, J.M., Jimenez, F., **A genetic algorithm for the Multi-objective solid transportation problem: A fuzzy approach.** 27th ISATA. Proceedings for the

Dedicated Conferences on Mechatorincs and Supercomputing Applications in Transportation Industries, (1994), PP: 327-34, ISBN: 0947719687

67. Jimenez, F., Verdegay, J.L., **Interval Multi-objective solid transportation problem via genetic algorithms.** (1996), hhtp://citeseer.ist.psu.edu/155465.html.

68. Pedgen, C.D., Shannon, R.E., Sadowski, R.P., ***Introduction to simulation using SIMAN.*** Second Edition ed. McGraw-Hill International, (1991), New York, ISBN: 0070492174.

69. Carson. J.S., Manivanan, M.s., Brazier, M., Miller, E., Ratliff, H.D., **PANEL ON TRANSPORTATION AND LOGISTICS MODELLING.** Proceedings of the Winter Simulation Conference, (1997).

70. Nilsson, F., Waidringer, J., **Toward adaptive logistics management.** Proceedings of the 38th Hawaii International Conference on System Sciences, (2005).

71. Davidson, I., Kowalczyk, R., **Towards better approaches to decision support in logistics problems.** CSIRO Mathematical and Information Sciences, (1994), http://citeseer.ist.psu.edu/666522.html.

72. Hamzacebi, C., Kutay, F., **A heuristic approach for finding the global minimum: Adaptive random search technique.** Applied Mathematics and Computation, Vol. 173, (2006), PP: 1323-33.

73. Holland, J.H., **Adaptation *in natural and artificial systems : An introductory analysis with applications to biology, control, and artificial intelligence.*** Ann Arbor: The University of Michigan Press, (1975), ISBN: 0472084607.

74. Goldberg, D.E., *Genetic algorithms in search, optimisation, and machine learning.* Addison Wesley Longman, Inc, (1953), USA, ISBN: 0-201-15767-5.

75. Michalewicz, Z., ***Genetic algorithms + data structures = evolution programs.*** 3rd. ed. Springer-Verlag, (1996), Berlin, New York, ISBN: 3540606769.

76. De Jonge, K.A., **Genetic Algorithms: A 10 year perspective, in proceeding of the first international conference on genetic algorithms.** Lawrance Erlbaum Associateds, (1985).

77. Reeves, C.R., ***Modern heuristic techniques for combinatorial problems.*** Oxford: New York: Blackwell Scientific; Halstead Press, (1993), London, ISBN: 0632032383.

78. Whitley, D., **The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best.** Proceeding of the third International Conference on Genetic algorithms, (1989).

79. De Jong KA. **An analysis of the behaviour of a class of genetic adaptive systems** (dissertation). University of Michigan, 1975.

80. Goldberg, D.E., Deb, K., Korb, B., **Do not worry, be messy.** Proceeding of the fourth international conference on Genetic algorithms, (1991), PP: 24-30.

81. Goldberg, D.E., Deb, K., **A comparative analysis of selection schemes used in genetic algorithms.** In G.J.E. Rawlins, editor, Foundations of Genetic algorithms, (1991), PP: 69-93.

82. Frantz, D.R., **Non-linearities in genetic adaptive search** (dissertation). USA: University of Michigan, MI.

83. Cavicchio, DJ. **Adaptive search using simulated evolution.** [Dissertation]. USA: University of Michigan MI; (1970).

84. Goldberg, D.E., Lingle, R., **Alleles, loci, and the travelling salesman problem.** Proceedings of an International Conference on genetic algorithms and Their Applications, (1985), PP: 154-9.

85. Davis, L., ***Handbook of genetic algorithms.*** Van Nostrand Reinhold, (1991), New York.

86. Harik, G., **Finding multimodal solutions using restricted tournament selection.** Proceeding of the 6th International Conference on Genetic algorithms, Eshelman, L.J. (Ed.), (1995).

87. Mahfoud, S.W., **Crowding and pre selection revised. .** Parallel Problem Solving from Nature 2, R. Manner, B. Manderick (Eds.), (1992), PP: 27-36.

88. Thierens, D., **Selection schemes, elitist recombination, and selection intensity.** Proceeding of the 7th International Conference on Genetic algorithms, T. Back (Ed.), (1997), PP: 152-9.

89. Wiese, K., Goodwin, S.D., **Convergence characteristics of keep best reproduction.** Selected Areas in Cryptography, (1999), PP: 312-8.

90. Ghosh, A., Tsutsui,H., Tanaka,H., **Function optimisation in nonstationary environment using steady state genetic algorithms with aging of individuals.** Proceeding of the IEEE International conference on Evolutionary Computation, (1998), PP: 666-71.

91. Vavak, F., **Genetic algorithm based self-adaptive techniques for direct load balancing in nonstationary environments.** [Dissertation]. University of the west of England, (1997).

92. Deb, K., Goel, T., **Controlled elitist non-dominated sorting genetic algorithms for better convergence.** Evolutionary Multi-Criterion Optimisation. First International Conference, EMO2001. Proceedings, (2001), PP: 67-81.

93. Ahn, C.W., Ramakrishna, R.S., **Elitism-based compact genetic algorithms.** IEEE Transaction on Evolutionary Computation, Vol. 7, Issue 4, (2003).

94. Laumanns, M., Zitzler, E., Thiele, L., **A unified model for multi-objective evolutionary algorithms with elitism.** Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), Vol. 1, Issue 1, (2000), PP:46-53, ISBN: 0 7803 6375 2.

95. Schaffer, J.D., **Multiple objective optimisations with vector evaluated genetic algorithms.** Proc of 1st ICGA, (1985), PP: 93-100.

96. Yang, Q.W., Jiang, J. P., Chen, G., **How to select optimal control parameters for genetic algorithms.** EEE International Symposium on Industrial Electronics, 2000 IEEE International Symposium on Industrial Electronics (ISIE'2000), Vol. 1, (2000), PP:37-41, .

97. Dejong, K.A., Spears, W.M., **An analysis of the interacting roles of population size and crossover in genetic algorithms.** Proceeding of the fifth Workshop parallel problem solving nature, (1990), PP: 38-47.

98. Grefenstette, J.J., **Optimisation of control parameters for genetic algorithms.** IEEE Transaction on Syst , Man, Cybern, Vol. 16, Issue 1, (1986), PP:122-8.

99. Hongwei, Z., Liangting, Y., Buyun, W., Gang, C., Haiping, Y., **A new genetic algorithm for unit commitment.** IEEE International Conference on Intelligent Processing Systems, (1997), PP: 606-10.

100. Brown, B.E., Gupta, S., Christie, R.D., Venkata, S.S., **A genetic algorithm for reliable distribution system design.** ISAP '96. International Conference on Intelligent.

101. Systems Applications to Power Systems Proceedings (Cat. No.96TH8152), (1966), PP: 29-33, ISBN: 0 7803 3115 X.

102. Hitchcock, F.L., **The distribution of a product from several sources to numerous locations.** Journal or Mathematics and Physics, Vol. 30, (1941).

103. Vignaux, G.A., Michalewicz, Z., **A genetic algorithm for the linear transportation problem.** IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, Issue 2, (1991), PP: 445-52.

104. Elmahi, I., Merzouk, S., Grunder, O., Elmoudni, A., **A genetic algorithm approach for the batches delivery optimisation in a supply chain.** Proceedings of the 2004

IEEE International Conference on Networking, Sensing and Control, (2004), PP:21-23, ISBN: 0-7803-8193-9/04/.

105.Jih, W.R., Hsu, Y.J., **Dynamic vehicle routing using hybrid genetic algorithms.** Proceedings of the IEEE International Conference on Robotics & Automation, (1999).

106.Garcia, J.M., Lozano, S., Smith, K., Kwok, T., Villa, G., **Coordinated scheduling of production and delivery from multiple plants and with time windows using genetic algorithms.** Proceeding of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 3, (2002).

107.Malmborg, C.J., **Theory and methodology: A genetic algorithm for service level based vehicle scheduling.** European Journal of Operational Research, Vol. 93,(1996), PP:121-34.

108.Park, Y.B., **A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines.** International Journal of Production Economics, Vol. 73, (2001), PP: 175-88.

109.Potter, T., Bossomaier, T., **Solving vehicle routing problems with genetic algorithms.** IEEE International Conference on Evolutionary Computation (Cat. No.95TH8099), Vol. 2, Issue 2, (1995), PP: 788-93, ISBN: 0 7803 2759 4.

110.Ochi, L.S., Vianna, D.S., Drummond, L.M.A., Victor, A.O., **A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet.** LECTURE NOTES IN COMPUTER SCIENCE, Issue 1388, (1998), PP: 216-24, ISBN: 0302-9743.

111.Baker, B.M., Ayechew, M.A., **A genetic algorithm for the vehicle routing problem.** Computers and Operations Research, Vol. 30, (2003), PP:787-800.

112.Wen, C., Eberhart, R.C., **Genetic algorithm for logistics scheduling problem.** Proceeding of the 2002 Congress on Evolutionary Computation CEC'02 (Cat. 02 TH8600), Vol. 1, Issue 1, (2002), PP: 512-6.

113. Murata, T., Ishibuchi, H., **MOGA: Multi-objective genetic algorithms.** 1995 IEEE International Conference on Evolutionary Computation (Cat. No.95TH8099), Vol. 1, Issue 1, (1995), PP: 289-94, ISBN: 0 7803 2759 4.

114. Horn, J., Nafpliotis, N., Goldberg, D.E., **A Niche Pareto genetic algorithm for multi-objective optimization.** Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence (Cat. No.94TH0650-2), Vol. 1, Issue 1, (1994), PP: 82-7, ISBN: 0 7803 1899 4.

115. **Surface transportation efficiency analysis model (STEAM)** [homepage on the Internet]. U.S. Department of Transportation Federal Highway Administration. May 25, 2005. Available from: http://www.fhwa.dot.gov/steam/.

116. **The capture and sequestration of carbon dioxide** [homepage on the Internet]. The Energy Systems Research Unit (ESRU), a member of the Institute for Energy and Environment at the University of Strathclyde. Available from: http://www.esru.strath.ac.uk.

117. **Traffic congestion and reliability: trends and advanced strategies for congestion mitigation, July 2004 report - traffic congestion and reliability: Linking solutions to problems** [homepage on the Internet]. Cambridge Systematic, Inc.: Texas Transportation Institute. 2005 September 1, (2005). Available from: http://www.ops.fhwa.dot.gov/congestion_report/index.htm.

118. **The costs of transportation: Expenditures on surface transportation the central budget found for 1995** [homepage on the Internet]. Washington State Department of Transportation. 1995. Available from: http://www.wsdot.wa.gov/Research/Reports/.

119. Litman, D., ***Characteristics of urban transportation systems.*** Cambridge Systematic, (1992).

120. **Surface transportation efficiency analysis model (STEAM)** [homepage on the Internet]. U.S. Department of Transportation Federal Highway Administration. May 25, 2005. Available from: http://www.fhwa.dot.gov/steam/.

121. Schriber, T.J., **The nature and role of simulation in the design of manufacturing systems.** Simulation in CIM and Artificial Intelligence Techniques, Society for Computer Simulation, (1987), PP: 5-18.

122. Law, A.M., Kelton, W.D., ***Simulation modelling and analysis.*** 3rd ed. McGraw-Hill Book Co, (2000), Singapore, ISBN: 0-07-116537-1.

123.Gupta, U.G., **Using citation analysis to explore the intellectual base, knowledge dissemination, and research impact on interfaces (1970-1992),** Vol. 27, (1997), PP:85-101.

124.Anu, M., **Introduction to modelling and simulation.** Proceeding of the 1997 Winter Simulation Conference, Eds, Andradottir, K.J. Healy, D.H. Withers, and B.L. Nelson, (1997), PP: 7-13.

125.Qiao, G., Lu, R.F., McLean, C., **Process control and logistics management for mass customisation manufacturing.** Proceedings of the IIE 2004 Annual Conference, Houston, TX, (2004), PP: 1-6.

126.Massei, M., Simeoni, A., **Application of simulation to small enterprise management & logistics.** Proceedings of the 2003 Summer Computer Simulation Conference, (2003), PP: 829-34, ISBN: 1-56555-268-7, http://www.scs.org/getDoc.cfm?id=2355.

127.Dong, M., Chen, F.F., **Performance modelling and analysis of integrated logistics chains: An analytic framework.** European Journal of Operation Research, Vol. 162,(2005), PP:83-98, WWW.sciencedirect.COM.

128.Dong, M., **Process modelling, performance analysis and configuration simulation in integrated supply chain network design.** Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University for the degree of Doctor of Philosophy, (2001), http://scholar.lib.vt.edu/theses/available/etd-08242001-102340/.

129.Chwif, L., Barretto, M.R.P., Saliby, E., **Supply chain analysis: Spreadsheet or simulation?.** Proceeding of the 2002 Winter Simulation Conference, E.Yucesan, C.H.Chen, J.L. Snowdon, and J.M. Charnes, eds., (2002), PP:59-66.

130.Joines, J.A., Gupta, D., Gokce, M.A., King, R.E., Kay, M.G., **Supply chain multi-objective simulation optimisation.** (2002), PP: 1306-14.

131.Schunk, D.P.B., **Using simulation to analyse supply chains.** (2000), PP: 1095-100.

132.Terzi, S., Cavalieri, S., **Simulation in the supply chain context: A survey.** Computers in Industry, Vol. 53, (2004), PP: 3-16, www.sciencedirect.com; www.elsevier.com/locate/compind.

133.Truong, T.H., Azadivar, F., **Simulation based optimisation for supply chain configuration design.** Proceedings of the 2003 Winter Simulation Conference, S. Chick, P.J Sanchez, D. Ferrin, and D.J. Morrice, eds., (2003), PP:1268-75.

134.Belhau,T., Strothotte,C., Ziems,D., Schurholz,A., Schmitz,M. ***Modelling and simulation of supply chain.*** (1999), www.uni-magdburg.de.

135.Rossetti, M.D., Chan, H.T., **A prototype object-oriented supply chain simulation framework.** Proceeding of the 2003 Winter Simulation Conference, S. Chick, P.J. Sanchez, D.Ferrin, and D.J. Morrice, Eds. , (2003), PP: 1612-20.

136.Dalal, M.A., Bell, H., Denzien, M., Keller, M.P., **Initinalizing a distribution supply chain simulation with live data.** Proceeding of the 2003 Winter Simulation Conference, S. Chick, P.J. Sanchez, D.Ferrin, and D.J. Morrice, eds., (2003), PP:1621-6.

137.Ramakrishnan, S., Lee, S., Wysk, R.A., **Implementation of a simulation-based control architecture for supply chain interactions.** Proceeding of the 2002 Winter Simulation Conference, E. Yucesan, C.-H. Chen, J.L. Snowdon, and J.M. Charnes, eds., (2002), PP:1667-74.

138.**Supply chain guru** [homepage on the Internet]. Available from: www.promodel.com.

139.Banks, J., Buckley, S., Jain, S., Lendermann, P., **Panel session: Opportunities for simulation in supply chain management.** Proceeding of the 2002 Winter Simulation Conference, E. Yucesan, C.H. Chen, J.L. Snowdon, and J.M. Charnes, eds., (2002), PP:1652-8.

140.Siprelle, A.J., Parsons, D.J., Clark, R.J., **Benefits of using a supply chain simulation tool to study inventory allocation.** Proceedings of the 2003 Winter Simulation Conference, S. Chick, P.J Sanchez, D.Ferrin, and D.J. Morrice, eds. , (2003), PP: 238-45.

141.Bruzzone, A., Orsoni, A., **AI and simulation-based techniques for the assessment of supply chain logistics performance.** Proceedings 36th Annual Simulation Symposium (ANSS-36 2003), (2003), PP: 154-64, ISBN: 0 7695 1911 3.

142.Nurmilaakso, J.M., **Supply chain scheduling using distributed parallel simulation.** Journal of Manufacturing Technology Management, Vol. 15, Issue 8, (2004), PP: 756-70.

143.Siprelle, A.J., Parsons, D.J., Phelps, R.A., **SDI industry pro: Simulation for enterprise wide problem solving.** Proceeding of the 1999 Winter Simulation Conference, (1999).

144.Narayanan, S.G.S., Srinivasan, K., **Simulation based decision support for supply chain logistics.** Proceeding of the 2003 Winter Simulation Conference, S. Chick, P.J. Sanchez, D.Ferrin, and D.J. Morrice, Eds. , (2003), PP: 1013-20.

145.Fu, M., **Optimisation for simulation: Theory and practice.** INFORMS Journal on Computing, Vol. 14, Issue 3, (2002), PP: 192-215.

146.April, J., Glover, F., Kelly, J.P., Laguna, M., **Practical introduction to simulation optimisation.** Proceeding of the 2003 Winter Simulation Conference, S. Chick, P.J. Sanchez, D.Ferrin, and D.J. Morrice, Eds., (2003), PP:71-8.

147.Andradottir, S., **A review of simulation optimisation techniques.** In Proceeding of the 1998 Winter Simulation Conference, D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan (eds.), (1998), PP: 151-8.

148.Laguna, M., Marti, R., **Neural network prediction in a system for optimisation simulations.** IEE Transactions, Vol. 34, Issue 3, (2002), PP: 273-82.

149.Banks, J., Carson, J.S., **Introduction to discrete-event simulation.** Proceeding of the 1986 Winter Simulation Conference, J.Wilson, J.Henriksen, S. Roberts (eds.), (1986), PP: 17-23.

150.Walters, S.A., **A practical objected-oriented approach for distributed real-time simulation.** AIAA/IEEE Digital Avionics Systems Conference. 12th DASC (Cat. No.93CH3327-4), (1993), PP: 135-40, ISBN: 0 7803 1343 7.

151.Bruzzone, A.G., Giribone, P., **Decision-support systems and simulation for logistics: Moving forward for a distributed, real-time, interactive simulation environment.** Proceedings 31st Annual Simulation Symposium (Cat. No.98TB100226), (1998), PP: 17-24, ISBN: 0 8186 8418 6.

152.Narayanan, S., Cowgill, J., Malu, P., Nandha, H., Patel, C., Schneider, N., Tendulkar, J., Carrico, T.M., DiPasquale, J., **Web-based distributed interactive simulation using java.** 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation (Cat. No.97CH36088-5), Vol. 3, Issue 3, (1997), PP: 2690-5, ISBN: 0 7803 4053 1.

153.Muller, H.J., **Towards agent systems engineering.** Data & Knowledge Engineering, Vol. 23, Issue 3, (1997), PP: 217-45, ISBN: 0169-023X.

154.Barbuceanu, M., Teigen, R., Fox, M.S., **Agent based design and simulation of supply chain systems.** Proceedings Sixth IEEE Workshops on Enabling Technologies:

Infrastructure for Collaborative Enterprises (Cat. No.97TB100178), (1997), PP: 36-41, ISBN: 0 8186 7967 0.

155.Pathak, S.D., Nordstrom, G., Kurokawa, S., **Modeling of supply chain: A multi-agent approach.** SMC 2000 Conference Proceedings. IEEE International Conference on Systems, Man and Cybernetics. Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions' (Cat. No.00CH37166), Vol. 3, Issue 3, (2000), PP:2051-6, ISBN: 0 7803 6583 6.

156.Bazzan, A.L.C., Klugl, F., **Guest editorial: Agents in traffic and transportation: Exploring autonomy in logistics, management, simulation, and cooperative driving.** Transportation Research Part C: Emerging Technologies, v 13, n 4, August, 2005, Agents in Traffic and Transportation: Exploring Autonomy in Logistics, Management, Simulations, and Cooperative Driving, p 251-254, Vol. 13, Issue 4, (2005), PP: 251-4, ISBN: 0968-090X.

157.Harrell, C., Tumay, K., **Simulation made easy: A manager's guide.** Engineering & Management Press, (1997), ISBN: 0898061369.

158.**BestFit** (homepage on the Internet). Palisade cooperation. 2007. Available from: http://www.palisade-europe.com/products.asp.

159.Naylor, T.H., Finger, J.M., **Verification of computer simulation models.** Management Science, Vol. 14, (1976), PP: 92-101.

160.Van Horn, R.L., **Validation of simulation results.** Management Science, Vol. 17, (1971), PP:247-58.

161.Welch, B.L., **The significance of the difference between two means when the population variances are unequal.** Biometrika, Vol. 25, (1938), PP: 350-62.

162.Taha, H.A., ***Operation research an introduction.*** 6th ed. Prentice-Hall, Inc., (1997), Upper Saddle River, New Jersey, USA, ISBN: 0-13281172-3.

163.Elegbede, C., Adjallah, K., **Availability allocation to repairable systems with genetic algorithms: A multi-objective formulation.** Reliability Engineering and System Safety, Vol. 82,(2003), PP:319-30.

164.Igbaria, M., Sprague Jr., R.H., Basnet, C., Foulds, L., **Case study: The impact and benefits of a DSS: The case of FleetManager.** Information and Managment, Vol. 31,(1996), PP:215-25.

165. Brindle A., **Genetic algorithms for function optimisation** (dissertation). USA: University of Alberta, Edmonton; 1981.

166. Baker, J.E., **Reducing bias and efficiency in the selection algorithm.** Proceeding of the second International Conference On Genetic algorithms, (1987), PP: 14-21.

167. Tsutsui, S., Ghosh, A., **A study on the effect of multi-parent recombination in real coded genetic algorithms.** Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, (1998), PP: 828-33.

168. Hong, T.P., Wang, H.S., Lin, W.Y., Lee, W.Y., **Evolution of appropriate crossover and mutation operators in a genetic process.** Applied Intelligence, Vol. 16, (2002), PP: 7-17.

169. Lin, W.Y., Lee, W.Y., Pei Hong, T., **On self-adaptive multi-population genetic algorithms.** IEEE International Conference on Systems, Man and Cybernetics. Conference Proceedings (Cat. No.02CH37349), Vol. 6, Issue 6, (2002), PP: 6, ISBN: 0 7803 7437 1.

170. The multipopulation genetic algorithm: Local selection and migration [homepage on the Internet]. Available from: http://www.pohlheim.com/Papers/mpga_gal95/gal2_1.html.

171. Kazerony, M., Lohmann, R., **Application of evolution strategy in parallel populations.** Proceedings of Parallel Problem Solving from nature, (1991), PP: 198-208.

172. Voigt, H.M., Santibanez-koerf,I., Born, J. ,**Hierarchically structured distributed genetic algorithm .** Proceeding of Parallel Problem Solving from Nature 2, (1992), PP: 145-54.

List of Publications

1. S.M.S. Khanian, T. Szecsi and M.S.J. Hashmi, Development of a decision support tool for vehicle routing and scheduling, in Proceedings of the 20th International Manufacturing Conference IMC20 - Knowledge Driven Manufacturing. (IMC-20), September 2003, Ireland.

2. S.M.S. Khanian, T. Szecsi and M.S.J. Hashmi, A Decision Support System for Delivery Scheduling, in Proceeding of the 3rd International Conference on Advanced Manufacturing Technology. (ICAMT 2004), 2004, Malaysia.

3. S.M.S. Khanian, T. Szecsi and M.S.J. Hashmi, A hybrid approach for vehicle routing and scheduling, ", in Proceedings of the 5th International Symposium on Soft Computing for Industry (ISSCI 2004) at the 5 th World Automation Congress (WAC-2004), June 28 - July 4, 2004, Spain.

4. S.M.S. Khanian, T. Szecsi and M.S.J. Hashmi, Evaluation and Comparison of Transportation Schedules, in proceeding of the 14th International Conference on Flexible Automation and Intelligent Manufacturing, (FAIM 2004), July 12-14, 2004, Toronto, Canada.

5. S.M.S. Khanian, T. Szecsi and M.S.J. Hashmi, A Genetic Based Decision Support System for Multicriteria Optimisation in Logistic Management, in proceedings of The 37th International Conference on Computers and Industrial Engineering, (ICCIE 2007) October (2007), Egypt.

Appendix A: Literature Survey

A.1 Integer Programming (IP):

The following sections describe different approaches for solving integer-programming problems.

A.1.1 Enumerative techniques

This approach, which involves enumerating all possibilities, is the simplest method to solve integer-programming problems. However, due to the large solution space resulting from the parameter size, only the smallest instances can be solved by such an approach. Usually constraints can implicitly eliminate many possibilities. The most commonly used enumerative approach is called branch and bound enumeration [A-1] where branching refers to the enumeration part of the solution technique and bounding refers to the grouping of possible solutions by comparison to a known upper or lower bound on the solution value.

A.1.2 Relaxation and decomposition techniques

An alternative approach to the solution of integer programming problems is to take a set of complicating constraints into the objective function. This method is known as Lagrangian relaxation [A-2]. This involves removing the complicating constraints from the constraint set, the resulting sub problem is considerably easier to solve. The resulting sub problem must be solved repetitively until optimal values for multiplies are found. The bound found by Lagrangian relaxation can be tighter than that found by linear programming, but only at the expense of solving sub problems in integers. This method requires that one understands the structure of the problem being solved in order to relax the constraints that are complicating the problem. In addition, the Lagrangian decomposition, involves isolating sets of constraints so as one can obtain separate, easy problems to solve for each of the subsets.

A.1.3 Cutting planes approaches

This approach was first introduced by [A-3]. The method assumes that all the data is rational. The implications of this are that the fractions can be cleared so that all constrains have integer coefficients. New constraints called cuts are introduced into the problem one at a time which progressively removes non-integer parts of the feasible region. This method is described as follows [A-1]:

This technique starts by first dropping the integer conditions and solving the resulting continuous linear program. By using the properties of the optimum continuous tableau, a constraint can be constructed which slices off part of the solution space such that no feasible integer solution is ever deleted. The continuous linear program is then solved subject to the new solution space. If the new optimum is integer, the process ends, otherwise a new cutting plane is constructed from the new tableau and reapplied to slice off another part of the new continuous space. The end effect of generating and applying these special constraints is that at the optimum extreme point of the modified solution space should satisfy the integrality conditions of the discrete variables. In this approach one cannot really tell in advance how many cuts should be generated before the integer conditions are realised.

A.2 Design Strategies for heuristic algorithms

The following sections briefly explain about four basic strategies used to establish heuristic procedures. These are taken from [A-4].

A.2.1 The construction strategy

The input for methods based on this strategy is the data, which defines a specific instance of the considered problem. This method begins by examining data and attempting to identify an element of the final solution, which is likely to be the part of the final solution. Next, successive additional elements of a solution are added. Also, some construction heuristics use some kind of "look-ahead" mechanism. That is, additions are made because they are likely to be a genuine value in the complete solution. Once the final solution has been built up it may be obvious that improvements can be easily effected. This strategy is worthwhile, when it is relatively difficult to generate feasible solutions to the problem.

A.2.2 The Improvement strategy

The input to the methods based on this strategy is a solution to the problem. This solution is then progressively improved by a series of modifications. In some instances, it may be impossible to make much more progress and yet the final product may still be far from optimal. This strategy is useful when it is relatively easy to generate starting solutions. A variety of solutions can be used as input and the best final result chosen, also, this strategy could be used to convert an infeasible solution into a feasible one.

A.2.3 The component analysis strategy

In general, there are some problems that are so large or so complicated that the only practical approach is to break them up into manageable portions. Sometimes these portions are then dealt with independently by heuristics or algorithms. The solutions for the portions are then joined to form some master plan. It could be very difficult to join together the solutions to different components into an acceptable plan.

A.2.4 The learning strategy

Methods based on this strategy often use a tree-search diagram to chart their progress. That is the different options, which appear at various stages are represented by different branches of a tree. The sequence of choices made can be traced typically using a path through the tree. The choice of which branch to take is guided by learning from the outcome of earlier decisions. The branch and bound search is an example of this strategy.

A.3 General Purpose Heuristics

The following sections provides a number of heuristic methods [A-5]:

A.3.1 Gradient Methods

A number of different methods for optimising well-behaved continuous functions have been developed which rely on using information about the gradient of the function to guide the direction of search. The gradient measures the rate and direction of change in a scalar field. The gradient is simply the derivative of a real valued function of a single variable or it is the slope of a linear function. These methods would often fail when the derivative of the function can't be calculated due to the discontinuity of the function. The Steepest decent is a method of this type that approaches a local maximum or a function by taking steps proportional to the gradient of the function at the current point. If steps are taken proportional to the negative of the gradient then a local minimum of that function is approached. Also, Hill climbing is a typical optimisation method of this type. This method can perform well with only one pick or uni-modal functions. However with multi-modal functions they suffer from the top of local maximum and no further progress can be made.

A.3.2 Random Search

Random search methods are variations on "rolling the dice" and are therefore called "Monte Carlo". The random search routines are usually used for optimisation of complex designs. Random search algorithms have achieved increasing popularity as researches have recognised the shortcomings of calculus-based and enumerative schemes. This method simply involves generating a random number or set of random numbers and using them for design. In general random search can handle mixed continuous/integer variable situations and they are not usually troubled by local optima. However, random walks and schemes must be discounted because of their lack of efficiency. These methods in the long run can be expected to do no better than enumerative schemes. The genetic **algorithms**imulation annealing are examples of methods that use random process to exploit the search space.

A.3.3 Iterated Local Search (ILS)

The underlying idea of an iterated local search is to build a random walk in the search space of the local optima generated by the output of a given local search algorithm. This technique has the advantage of simplicity, and can perform well if the function does not have too many local maxima. The effectiveness of the local search is important as it influences the quality of the final solution and the computation time. Typically, random search and gradient search may be combined to provide an iterated hill-climbing search method. In this approach once a pick has been specified then the local search (hill-climb) is started again, but with another, randomly chosen starting point.

The random search is used as a perturbation mechanism, which allows the local search to effectively escape local optima. However, since the random search is carried out in isolation, no overall picture of the shape of the domain is obtained. As the random search progresses, it continuous to allocate its trials evenly over the search space. Therefore, in this way there would be as many points in regions found to be of low fitness as in regions found to be of high fitness.

A.4 Separate and Single Origin and destination points:

The followings present a number of problems used to describe separate and single origin and destination points case as one of the main category of VRPs.

- **The Shortest Path Problem:** This is a network flow problem that is important from applied standpoint. The problem here is to determine a path from the source node to the sink node, which minimises the total cost or time spent in the shipment of a given amount of flow along that path. All algorithms addressing the shortest path problems are based upon the following observation [A-6]:

"if the shortest route between node S and t passes through node k, then that segment of the route from node s to node k is also the shortest route to mode k. in addition, the route from node k to node t is the shortest route between those two nodes".

- **Shortest-Path Models with Fixed Charges:** Fixed charge problems arise when certain additional penalties or costs are incurred in traversing through one or more nodes in a network. In general such problems occur when transhipment facilities must be built, bought, or rented for interim usage. As opposes to the algorithms in the shortest path problems, in a network with turn penalties, *the shortest route from node s to node t through an intermediate node k may not include the shortest route from node s to node k, or form node k to nose s.*
- **The K-Shortest Path Problem:** It is sometimes desirable to have knowledge of several shortest paths, ranked according to their lengths, in contrast to simply a shortest path as preformed in the latter approaches. As an example, transportations planners could use alternative solutions to model the flow of vehicles in a road network even more realistically. Therefore, identification of additional solutions provides and alternative approach for planning when the vest solution is not available or in infeasible.
- **Minimal Spanning Tree Problem:** This problem is found to be useful in many real world applications. For instance, considering a supply source of a natural gas and there is a need to supply a number of distinct demand centres from this raw martial supply. The minimal spanning tree for the supply network would be that distribution system that would connect all users in a minimum total cost (or distance). Minimal spanning trees are also often used in sub-optimisation or decomposition of larger, more complex network algorithms. In general, a tree is a set of connected undirected edges (arcs) that contains no cycles. As a result, a given set of ***m*** nodes connected by undirected edges, a subset of exactly ***[m-1]*** arcs is needed to form a tree. In this way each node is connected to another node by a unique path.

A.5 Coincident Origin and Destination Points

The following problems are used to describe these categories of VRP [A-7].

- **The Travelling Salesman Problem (TSP)**: This requires the determination of a minimal cost cycle that passes through each node in the relevant graph exactly once. If costs are symmetric, which is the cost of travelling between two locations does not depend on the travelling direction, then this is known as symmetric travelling salesman problem; otherwise, it is considered as asymmetric or directed travelling salesman problem.
- **Chinese postman problem**: This requires determination of the minimal cost cycles that passes through every arc of the graph at lease one time. A Chinese postman problem is called directed or undirected, depending on whether arcs of the graph are directed or not. The mixed Chinese postman problem has some undirected and some directed arcs; this problem is NP-hard.
- **The Single Depot Multiple Vehicle Routing Problems**: This is the classical vehicle routing problem. This problem requires a set of delivery routes for vehicles housed at a central depot, which service all the nodes and minimise the total distance travelled. The demand at each node is assumed to be deterministic and each vehicle has a known capacity.
- **The Multiple Depot Multiple Vehicle Routing Problems:** This is a generalisation of the last problem in that the fleet of vehicles now must serve a number of depots rather

than just one. All other constraints from the classical VRP still apply. In addition, each vehicle must leave from and return to the same depot.

A.6 Vehicle Scheduling Problems (VRSP)

The following problems are used to describe Vehicle Scheduling Problems (VRSPS) [A-7]:

- **The single depot Vehicle Scheduling Problem (VSP):** This requires the partitioning of the nodes (tasks) into a set of paths in such a way that a certain cost function is minimised. Each path corresponds to the schedule for a single vehicle. An objective function that minimises the number of paths effectively minimises capital costs since the number of vehicles equals the number of paths. Also, if weight equals to the corresponding deadhead, then an objective function that minimises total arc weight effectively minimises operating costs since these are proportional to the total vehicle travel time. Finally, if capital and operating costs can be quantified, then a combined objective can be used to minimise total system costs.
- **Vehicle Scheduling Problem with Length of Path Restrictions (VSPLPR):** In this case, the constraints are placed on the length of time a vehicle may spend away from the depot or the mileage a vehicle may cover without returning to the depot for service. This constraint is used to represent fuel restrictions, maintenance considerations, etc. whereas the VSP can be solved using a polynomially- bounded algorithm, the VSPLPR is NP-hard.
- **Vehicle Scheduling Problem with Multiple Vehicle Types (VSPMVT):** This allows the possibility that vehicles with different characteristics are available to service the tasks. In most cases, the characteristic is vehicle capacity. For instance, in LPG transportation systems, a truck with small capacity can service the trips with low demand; regular trucks can serve trips with high demand and either vehicle can service trips with medium demand. Therefore, in this application, for each task the set of vehicles that may service it is specified.
- **Vehicle Scheduling Problem with Multiple Depots (VSPMD):** In this scenario, tasks may be serviced out of more than one depot. As with the vehicle routing problem, each vehicle must leave and return to the same depot and the fleet size at each depot must range between a specified minimum and maximum.

A.7 Vehicle Routing and Scheduling Problems (VRSPs)

The following demonstrates a typical vehicle routing and scheduling problem.

- **Dial-a-ride routing and scheduling Problems:** In the dial-a-ride problem, customers call a dispatcher or scheduler requesting service. Each customer specifies a distinct pickup and delivery point and perhaps a desired time for pickup or delivery. If all customers demand immediate service, then routing and scheduling is done in real time and the problem is referred to as the dynamic or real time dial-a-ride problem. If all customers call in advance, so that a complete database of customer demand is known before any routing or scheduling is carried out, then this problem is referred to as the subscriber or static dial-a-ride problem. Both the dynamic and static dial-a-ride problems have precedence relationships since a customer must be picked up before he is delivered. Dial-a-ride problems and their extensions occur in many applications shared cab rides, package delivery, bank deliveries, etc.

A.8 Heuristics for the VRPs

Table A- 1 Classical heuristics for single origin & destination points

<table>
<tr><th colspan="2">Separate Single Origin & Destination Points</th></tr>
<tr>
<td>

1. <u>Dijkstai Algorithm</u> [A-8]:

- The algorithm operates on the simple logic that if a shortest path from node S to node j is known and node k belongs to this path, then the minimal path from s to k is the portion of the original path ending at node k.
- The algorithmstart with j = s and successively resets j until j = t and then the process is stopped.

</td>
<td>

2. <u>Multiterminal shortest chain route</u> [A-9]:

- The shortest chain between any pair of nodes i & k is the length of the connecting arc [i, k].
- The algorithmsuccessively examines all possible intermediate nodes between i & k, and if the length of chain through any intermediate node is shorter than the current distance d_{ik}, then d_{ik} is changed to the new value.
- This procedure is repeated for all possible pairs of nodes until all d^{*}_{ik} values are obtained.
- <u>Number of computations required</u>: n iterations for network with n nodes.

</td>
</tr>
<tr>
<td>

3. <u>Modified shortest path Iterative</u> [A-6]:

- Add a pseudo node $\bar{S}$ and connect it to the source node s by direct arc [$\bar{S}$, s]; add a pseudo node $\bar{t}$ and connect the terminal node $\bar{t}$ to t by direct arc [$\bar{t}$, t].
- Assign a pseudo label l_k to each arc of the network. Let l_0 , l_1 ,…, $l_{\alpha+1}$ be the pseudo label for $\alpha + 2$ arcs .
- Create a pseudo network consisting of $\alpha + 2$ nodes l_0 , l_1 ,…, $l_{\alpha+1}$ such that the directed arc (l_i , l_j) in the pseudo network is defined if the arc with label l_i immediately proceeds the arc with label l_j in the labelled network of step 2.
- The arc parameter of the branch connecting node l_i to node l_j is given by c [l_i] +p (l_i , l_j) , where c [l_i] is the original cost of l_i and p (l_i , l_j) is the turn penalty associated with the branch (l_i , l_j) .
- Then use the shortest path to find the most economical path.
- Translate the found solution back into the original network formulation.

</td>
<td>

4. <u>Double Sweep Method</u> [A-10]:

- Assuming that for each node there is vector with estimates of the K shortest path length from a given source node. This approach successively reduces the estimates until the optimal vector of estimates is achieved in finite number of iterations.
- Each iteration consists of two passes.
- In the forward pass, the nodes are considered in increasing numerical order. After identifying the list of nodes i incident to node j, such that $i<j$, the k shortest path lengths from the source to node j are successively examined to verify if shorter path lengths are possible through the incident nodes.
- If such path lengths exist, they will be used as new estimates in further iteration.
- This is followed for the backward pass of the algorithm, but the nodes are considered in decreasing order and only nodes, where $i>j$ are investigated.

</td>
</tr>
</table>

Table A-2 Classical heuristics for multiple origin/destination points

<table>
<tr><th colspan="2">Multiple Origin & Destination Points</th></tr>
<tr>
<td>

2. <u>Simplex Method</u> [A-11]:

- Select a starting basic feasible solution, [using either northwest or VAM approaches]
- Check if the solution is improved by introducing a non-basic route into the basis.
- Determine which route must leave the basis when the route chosen in step 2 enters.
- Adjust the flows on the other basic routes to preserve the feasibility of the new basic solution.
- Go back to step 2, if the solution is improved go to step, else exit the procedure.

</td>
<td>

4. <u>Vogel Approximation Method [VAM]</u> [A-12]:

- Calculate the minimal penalty incurred if the most economical route is not chosen to leave each source or to go to each terminal, and
- Then select the source or terminal associated with the largest minimal penalty.
- Assign as many units of flow as possible to the most economical route leaving the source or going into the terminal, whichever was chosen.
- After the chosen route is saturated, the source and the terminal linked by the route are checked to see what supply-demand condition becomes satisfied.
- The source or terminal whose condition is satisfied is removed,
- New penalties are computed, and the procedure is

</td>
</tr>
</table>

	repeated until all sources and terminals are removed. • The penalty is equal to the difference between the smallest and the next-to-the-smallest costs. • If both costs are equal, the penalty is then equal to zero. • Also, when the largest minimal penalty is not unique for a set of sources and or terminal, the ties can be broken arbitrarily.
4. <u>Northwest Corner Method</u> [A-13]: • Assign as much flow as possible to the route linking the first source with the first terminal. • Let X_{11} be the number of units assigned. • If $a_1 = b_1$, delete both source 1 and terminal 1, and select the route corresponding to X_{22}. • Otherwise, delete the source or terminal whose supply or demand is satisfied, and select either X_{12} or X_{21} depending on which one corresponds to a route that is feasible. • Assign as much flow as possible to the selected route and remove the source or terminal whose supply or demand is also satisfied. • If the supply and the demand conditions are simultaneously satisfied, both the source and the terminal are removed. • If X_{ij} is the last basic variable selected, • the next one to be considered is $X_{i,j+1}$ If the *ith* source has any supply available, • $X_{i+1,j}$ If the *jth* terminal has any unfilled demand, • or $X_{i+1,j+1}$ otherwise. •	**3. <u>Hungarian Algorithm</u>** [A-14]: • **<u>Row-column reduction process:</u>** • Subtract the smallest value in each row from all the elements in that row, and • Then subtract the smallest element in each column from all the elements in that column. • Use this reduced-cost matrix for assignments. • **<u>Identification of assignments:</u>** • If the row-column reduction process allows us to select one zero-cost cell from each row and one from each column, such that the resulting solution is feasible, that assignment will also be optimal. • If a zero-cost assignment in the reduced matrix is not possible, the matrix must be further modified, • **<u>Modification of the reduced matrix:</u>** • If there are insufficient zero-cost cells to select a zero-cost assignment, more zeros can be created by • Identify the minimal group of rows and columns containing all the zero-cost entries of the reduced matrix, and find the minimal value not in this group. • If this value is subtracted from all the entries of the matrix, the zeros become negative, and at least one element outside the group becomes equal to zero. •

Table A-3, Classical Heuristics for TSP (VRP)

TSP Heuristics	
3. <u>Nearest Neighbour</u> [A-11]: • Start with any node at the beginning of a path. • Find the node closest to the last node added to the path. Add this node to the path. • Repeat until all nodes are connected. • <u>Number of computations required</u>: on the order of n^2.	**4. <u>Clark and Wright Savings</u>** [A-15] • Select any node as the central depot. • Compute saving $S_{ij} = C_{1i} + C_{1j} - C_{ij}$ • Order the savings from largest to smallest. • Starting from the top to down of the saving list, form a tour linking nodes. Repeat until the tour is formed. • <u>Number of computations required</u>: on the order of $n^2 \lg(n)$.
3. <u>Nearest merger</u> [A-11]: • When applied on *n* nodes, it constructs a sequence $S_1,\ldots,S_n$ such that each S_i is a set of *n-i +1* disjoint subtours covering all the nodes. At each step of this procedure two subtours closest to one another are merged. • <u>Number of computations required</u>: on the order of n^2.	**4. <u>Minimal Spanning Tree</u>** [A-13]: • Find a minimal spanning tree T of G. • Double the edges in the minimal spanning tree to obtain Euler cycle. • Remove polygons over the nodes with degree>2. • Transform the Euler to Hamiltonian cycle. • <u>Number of computations required:</u> on the order of n^2.

5. Insertion Procedures [A-11]: • Consists of two steps: • The Selection Step: Takes a subtour on K nodes at iteration k and attempts to determine those nodes not in the subtour to join the subtour & then • Insertion Step: Determines where in the subtour it should be inserted.	**5. Christofides Heuristic** [A-16]: • Find a minimal spanning tree T of G. • Identify all the odd degree nodes in T. • Solve a minimum cost on the odd degree nodes. • Add the branches fro the matching solution to the branches already in T. • Obtain an Euler Cycle. • Remove polygons over the nodes with degree>2. • Transform the Euler into Hamiltonian cycle.

Table A-4 Classical heuristics for Chinese Postman Problem (CPP)

Chinese Postman Problem	
4. Chinese Postman Problem Algorithm [A-7]: • Using a shortest path algorithm on the graph G with cost matrix [C_{ij}] from the $\|N_o\|$ matrix $D=\lfloor d_{ij} \rfloor$ where d_{ij} is the cost of the least-cost path from node $i \in N_o$ to another node $j \in N_o$. • Use a polynomial-time minimum 1-matching algorithm to find $\overline{A}\text{min}$ according to the cost matrix D. • If node α is matched to another node β, identify the least-cost path $\mu_{\alpha\beta}$ corresponding to the cost $d_{\alpha\beta}$ of step1. Insert artificial arcs to obtain $G_o(\overline{A}\text{min})$. • The sum of the costs from matrix [C_{ij}] of all arcs in $G_o(\overline{A}\text{min})$ is the minimum cost of a cycle covering G. • Once the graph $G_o(\overline{A}\text{min})$ is obtained, start at any node, • Traversing and then erasing an incident arc such that the removal of the arc does not divide the graph into two disjoint components. • Subsequent nodes are treated one at a time.	**6. Constrained based CPP Algorithm** [A-17]: • Let E_i be the variable representing $e_i \in E$, so that the values of E_i indicates where e_i comes in the ordering for service and let $Edge_{Depot}$ be the variable representing the arbitrary edge representing the depot node. • Let $Edge = \{E_i, Edge_{depot} \| i = 1,...N\}$ be the set of all model variables, then the domain of the edge variable will be {1,…, N+1} with a cardinality of N+1. • For each edge $e_i = (p,q)$, let Z_i be a [0,1] variable used to determine the direction that the edge is serviced. • Suppose $p<q$, if the edge is serviced in the direction $p \rightarrow q$ then Z_i=0 and if the edge is serviced in the direction $q \rightarrow p$ then Z_i=1. • In order to find the cost [or time] between finished service of edge e_i and starting to serve ej when ej follows e_i in the ordering, • The direction variables can be used to indicate the finishing node for e_i and the starting node for ej so that the appropriate shortest path cost can be used from pre-calculated arrays. • Any shortest path algorithm can be used to calculate the values for the costs between e_i and ej.

Table A-5, Classical heuristics used to address VRPs

Single/Multiple Depot, Multiple Vehicle Routing Heuristics [VRPs]	
1. The Saving Algorithm [A-18]: • This is an exchange procedure. • At each step one set of tours is exchanged for a better set. • Initially, it is assumed that a separate vehicle supplies every demand point individually. • Then the savings in travel distance that may obtain using one vehicle to service node *i* and *j* is calculated. • Other approaches [r6-14, r6-15] used the saving criteria of this procedure to address VRPs.	**2. A Generalised assignment Heuristics** [A-20]: • This procedure is composed of two interrelated components. • One component is TSP and the other is a generalised assignment problem. • The TSP and the generalised assignment problems have been studied extensively in the literatures. • Therefore this formulation of VRP can benefit from the theory and algorithms developed for these problems.
3. A penalty Algorithm [A-19]: • Provides explicit consideration of customer demand by moving capacity constraints into the objective function. • It also applies a multiplier in order to impose a penalty when demand on a route exceeds capacity. • This approach tends to provide better results [Bodin et al.].	**4. M-Tour Approach** [A-21]: • The M-tour algorithm requires a feasible solution to the VRP with M vehicles as input. • This M-tour solution is expressed as a travelling salesman tour on an expanded network. • Then a modified 3-opt or any other improvement procedure is used to reduce the total costs.

5. The Sweep Approach [A-22]: • First, nodes [i.e. customers] are assigned to vehicles, • The order in which each vehicle visits the nodes is sequenced. • A "Seed" node is selected randomly. • A ray is swept [+/- directions] to the seed from central depot. • Demand points are added to a route as they are swept. • When the route's capacity is reached, then • The additional demand node becomes the seed for the next route. • TSP algorithms can be use to improve routes.	**6. A multi-depot saving algorithm for large problems** A-23]: • First nodes are allocated to depots and then routes are built linking nodes assigned to the same depot. • For every node i, the closest depot K_1 and the second closest depot K_2 are determined. • If Ratio $r(i) = C_i^{K_1} / C_i^{K_2}$ is less than a certain chosen parameter [$(0 \le \delta \le 1)$, i is assigned to K_1, otherwise i is set as a border node. • The Multi-depot savings algorithm is applied to allocate the border nodes to depots & to build segments of routes connecting these nodes. • Using single depot VRP techniques, the border nodes are extended to the remaining nodes.
7. A Multi Depot Saving Approach [A-24]: • The initial solution starts by assigning a rout to each node from the closest depot. • The method successively links pairs of nodes to decrease the total cost. • Once two or more nodes have been assigned to a common route from a depot, the nodes are not assigned to another depot. • At each step, the choice of linking node i and j on a route from depot k is made based on the saving when linking i and j at k.	**8. The Assignment-Sweep Approach** [A-25]: • Solves Multi-depot problem in two stages. • First, locations are assigned to depots, then • For each node the closeness ratio *r[i]* to each depot is calculated. Nodes are ranked by increasing value of *r[i]*. • The arrangement determines the order that nodes are assigned to depots. • The sweep heuristic is used to construct & sequence routes in the cluster about each depot independently.

A.9 Heuristics for the VSP

Table A-6 Classical heuristics to address VSPs

Vehicle Scheduling Heuristics (VSP)	
1. The Concurrent Scheduler:[A-26]: • Tasks are ordered by their starting times. • If it is feasible to assign a task to an existing vehicle, then it is assigned to the vehicle with minimum deadhead time • Otherwise the task is assigned to a new vehicle. • It is possible to solve all the three possible categories of VSP using this approach.	**2. Two Steps Approaches** [A-7]: • The VSPMVT and VSPMD can be addressed using this approach. • There are two classes of scheduling in this method: • First, clustering tasks and then scheduling vehicles over each cluster or • First, scheduling vehicles and then clustering vehicle schedules.
3. An interchange heuristic:[A-26]: • It can handle a wide variety of cost functions & constraints. • Assuming that starting schedules is already available for vehicles. • The procedure interchanges between the components of the schedules to improve the costs. • The cost of the new schedule is compared with the old vehicle schedules.	

A.10 Heuristics for the VRSPs

Table A-7, Classical heuristics to address VRSPs [A-7]

VRSP Heuristics	
1. Saving Heuristic: • Initially introduced by Clarke and Wright [1964] is extended to incorporate time windows to address VRSPTW problems. • In addition to take into account vehicle capacity constraints, this approach checks for time window constraints for violation at every step. • The heuristic could find it profitable to join two costumers very close in distance but far apart in time.	• **A Time-oriented, nearest –Neighbour Heuristic:** • It starts every route by finding the un-routed customer closest to the depot. • In every step, the heuristic searches for the customer closest to the last customer added to the route. • The feasibility conditions are Time Windows, Vehicle arrival time at the depot, Capacity constraints.

3. **Insertion heuristics:**	4. **A Time-Oriented Sweep heuristic:**
• This approach initialises every route using one of several criteria such as: the farthest unrouted customer or the unrouted customer with the earliest due date. • The method uses two criteria $C_1(i, u, j)$, $C_2(i, u, j)$ to insert a new customer into the current partial route. • For each unrouted customer, the best feasible insertion place in the route is obtained, then • The best customer to be inserted in the route is selected. • The insertion of unrouted customers is guided by both geographical and temporal criteria.	• There are clustering stage and a scheduling stage in this approach. • First, customers are assigned to vehicles as in the original sweep heuristic [Gillet and Miller 1974], then • Using a tour building heuristic a one-vehicle schedule is set for this cluster. • Repeat the clustering-scheduling process for the unscheduled customers. • The sequencing aspect of the problem seems to drive routing problems dominated by time windows.

A.11 GA applications within VRSP

The following are some research conducted in the area of GA application in VRSP.

- **Jih et al**. [A-27] proposed a hybrid approach taking advantage of both dynamic programming and GA methods for vehicle routing problem. The dynamic programming is used to generate the optimal routes. If the optimal solutions are not found within the specified time slot, the partially constructed routes are passed to GAs. This hybrid approach enables dynamic programming to achieve real time performance and improve GAs in approximating near optimal solutions.
- **Elmahi et al.** [A-28] proposed a GA that provides the optimal or nearest optimal, sequence of shipments within a supply chain.
- **Malmborg** [A-29] used GA to address service levels based vehicle scheduling with the objective being to minimise the delay between the time that material accumulates at each work centre, and the time that it is delivered to its destination work centre.
- **Potter et al.** [A-30] applied a two level GA to a general pick up and delivery problem representing an advanced transportation problem. In this approach an upper level GA was used to allocate passengers to vehicles, and a lower level GA was used to find the shortest route for a given set of passengers in a single vehicle. This splitting allows addressing the lower problem independently. The results obtained in this work suggest the effectiveness of GAs in quickly obtaining good solutions to vehicle routing problems.
- **Ochi et al.** [A-31] indicates that the inclusion of the notation of scatter search, tabu search and local search in GA could make GAs more efficient. In addition, using parallel techniques could also improve the running length of GAs before coming up with a solution. In this work a new hybrid Meta-heuristic which uses parallel GA and scatter search procedures was used to solve vehicle routing problems with a heterogeneous fleet.
- **Chen Wen** [A-32] developed a GA method to address logistics scheduling problems. They suggest logistics scheduling are problems that must be tackled with a combination of search techniques and heuristics. In this approach they considered all constraints and requirements in the GA fitness function. This consideration is not possible using traditional linear algorithms. They defined two fitness metrics for one chromosome. One fitness measure is the number of items delivered and the other is the number of missions completed. Mission refers to accomplishment of deliveries within its specified time window.

A.12 GA application to address Multi objective problems

[A-33] proposed a GA based solution method for a case in which fuzzy goals are assumed in the multi-objective solid transportation problem. [A-34] proposed a GA method to address a multi-objective solid transportation problem. In this approach three criteria were considered concerning, supply, demand and conveyance capacity for the solid. As there is always uncertainty on these criteria, the developed GA is based on interval values for the main considering criteria. This solution method is known as a non-standard GA.

References A

A - 1. Land, A.H., Doig, A.G., **An automatic method for solving discrete programming problems.** Econometrica, Vol. 28, (1960), PP: 97-520.

A - 2. Hoffman, K., **Combinatorial and integer optimisation.** George Mason University, http://citeseer.ist.psu.edu/247224.html.

A - 3. Cohon, J.L., ***Multi-objective programming and planning.*** Academic Press, (1978), New York, ISBN: 0121783502.

A - 4. Kazerony, M., Lohmann, R., **Application of evolution strategy in parallel populations.** Proceedings of Parallel Problem Solving from nature, (1991), PP: 198-208.

A - 5. Kazerony, M., Lohmann, R., **Application of evolution strategy in parallel populations.** Proceedings of Parallel Problem Solving from nature, (1991), PP: 198-208.

A - 6. Philips, D.T., Diaz, A.G., ***Fundamentals of network analysis.*** Prentice-Hall, (1981), Englewood Cliffs, N.J, ISBN: 013341552X.

A - 7. Bodin, L., Golden, B., Assad, A., Ball, M., **Routing and scheduling of vehicles and crews, the state of the art.** An International Journal of Computers and Operations Research, Vol. 10, Issue 2, (1983), PP: 63-211.

A - 8. Dijkstra, E.W., **A note on two problems in connection with graphs.** Numerishe Mathematik, Vol. 1, (1959), PP: 269-71, http://www.garfield.library.upenn.edu/classics1983/A1983QA19900001.pdf.

A - 9. Dreyfus, S.E., **An Appraisal of some shortest path algorithms.** International Journal of Operations Research, Vol. 17, (1969), PP: 395-412.

A - 10. Shier, D.S., **Iterative methods for determining the K shortest paths in networks.** Networks, Vol. 6, (1976), PP: 205-30.

A - 11. Rosenkrantz, D., Sterns, R., Lewis, P., **An analysis of several heuristics for the travelling salesman problem.** SIAM Journal of Computers, (1977), PP: 581-653.

A - 12. 127. Mathirajan, M., Meenakshi, B., **Experimental analysis of some variants of VOGEL's approximation method.** Asia-Pacific Journal of Operational Research, Vol. 21, Issue 4, (2004), PP: 447-62.

A - 13. Kim, C., **A minimal spanning tree and approximate tour for travelling salesman.** Comp Sci Tech Report, University of Mary Land, (1975).

A - 14. Golden, B., **A statistical approach to the TSP.** Networks, Vol. 7,(1977), PP:209-25.

A - 15. Clark, G., Wright, J.W., **Scheduling of vehicles from central depot to a number of delivey points.** International Journal of Operations Research, Vol. 12,(1964), PP:568-81.

A - 16. Christofides, N., **Modern methods of vehicle scheduling.** Freight Managment, (November 1979), PP: 53-6.

A - 17. Aminu, U.F., Eglese, R.W., ***A constraint programming approach to the Chinese postman problem with time windows.*** Computer and Operations Research, Vol. 33, (2006), PP: 3423-32, www.sciencedirect.com.

A - 18. Carpaneto, G., Toth, P., **Some new branching and bounding criteria for the asymmetric travelling salesman problem.** International Journal of Management Science, Vol. 26, Issue 7, (1980), PP: 736-43.

A - 19. Stewart, W., **New algorithms for deterministic and stochastic vehicle routing problems.** Working Paper No.81-009, University of Maryland at College Park, (1981).

A - 20. Fisher, M., Jaikumar, R., **A generalised assignment heuristic for vehicle routing.** Networks, Vol. 11, Issue 2, (1981), PP: 109-24.

A - 21. Russell, R.A., **An efficient heuristic for the M-tour travelling salesman problem with some side conditions.** Operation Research, Vol. 25, (1977), PP: 517-24.

A - 22. Garfinkel, R., Nemhauser, G., **The set-partitioning problem: Set covering with equality constraints.** International Journal of Operations Research, Vol. 17, (1969), PP: 848-56.

A - 23. Golden, B., Magnanti, T.L., Nguyen, H., **Implementing vehicle routing algorithms.** Networks, Issue 7, (1977), PP: 113-48.

A - 24. Tillman, F., Cain, T., **An upper bounding algorithm for the single and multiple terminal delivery problem.** Management Science, Vol. 18, (1972), PP: 664-82.

A - 25. Gillett, B., Johnson, J., **Multi-terminal vehicle dispatch algorithm.** Omega, Vol. 4,(1976), PP:711-718, http://www.tdx.cesca.es/TESIS_UPC/AVAILABLE/TDX-1019104-175236//09Mas09de10.pdf.

A - 26. Bodin, L., Rosenfield, D., Kydes, A., **UCOST.A micro approach to a transit planning problem.** Journal of Urban Analysis, Vol. 5, Issue 1, (1978), PP: 47-69.

A - 27. Jih, W.R., Hsu, Y.J., **Dynamic vehicle routing using hybrid genetic algorithms.** Proceedings of the IEEE International Conference on Robotics & Automation, (1999).

A - 28. Elmahi, I., Merzouk, S., Grunder, O., Elmoudni, A., **A genetic algorithm approach for the batches delivery optimisation in a supply chain.** Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control, (2004), PP:21-23, ISBN: 0-7803-8193-9/04/.

A - 29. Malmborg, C.J., **Theory and methodology: A genetic algorithm for service level based vehicle scheduling.** European Journal of Operational Research, Vol. 93,(1996), PP:121-34.

A - 30. Potter, T., Bossomaier, T., **Solving vehicle routing problems with genetic algorithms.** IEEE International Conference on Evolutionary Computation (Cat. No.95TH8099), Vol. 2, Issue 2, (1995), PP: 788-93, ISBN: 0 7803 2759 4.

A - 31. Ochi, L.S., Vianna, D.D., Drummond, L.M.A., Victor, A.O., **A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet.** Future Generation Computer Systems, Vol. 14, Issue 5-6, (December 1998), PP: 285-92, ISBN: 0167-739X.

A - 32. Chen Wen, Eberhart, R.C., **Genetic algorithm for logistics scheduling problem.** Proceedings of the 2002 Congress on Evolutionary Computation, Vol. 1, Issue 1, (2002), PP: 12-6, ISBN: 0 7803 7282 4.

A - 33. Cadenas, J.M., Jimenez, F., **A genetic algorithm for the Multi-objective solid transportation problem: A fuzzy approach.** 27th ISATA. Proceedings for the Dedicated Conferences on Mechatorincs and Supercomputing Applications in Transportation Industries, (1994), PP: 327-34, ISBN: 0947719687.

A - 34. Jimenez, F., Verdegay, J.L., **Interval Multi-objective solid transportation problem via genetic algorithms.** (1996), hhtp://citeseer.ist.psu.edu/155465.html.

Appendix B: Methodology

B.1 Rank-based Selection

The nature of scaling procedures associated with the roulette wheel selection led [B-1] to consider a nonparametric procedure for selection. In this method the population is sorted according to objective function value. Chromosomes are then assigned a number dependent only on their respective position in the rank. This method would overcome the scaling problem related to premature convergence and stagnation and the size of gaps between fitness's become irrelevant. Since the reproductive range is limited, no chromosomes generate an excessive number of offspring. Ranking would introduce a uniform scaling across the population and provides. However, this method has been criticized for disassociating the fitness function from the underlying objective function and this procedure does not provide a consistent means of controlling offspring allocation [B-2].

B.2 Crossover Methods

The following present a number of popular crossover methods presented in literatures.

- <u>**Order Crossover (OX)**</u>: Proposed by [B-3]. This operator generates offspring by choosing a random fragment from one parent and preserving the relative order of the genes from the other parent. The following steps would be used to produce the offspring:
    - First the random fragments are copied into offspring (Figure B-1, Figure B-2),
    - Starting from the second cut point of one parent, the genes from the other parent are copied in the same order, removing elements that are already available.
    - When reaching the end of the string then the next position would be the first location of the string.

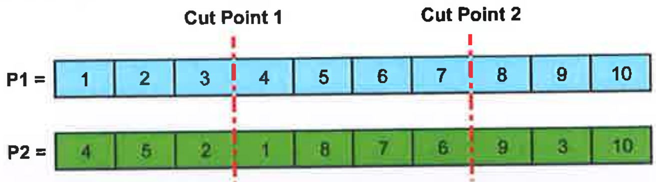


Figure B-1 Before OX Crossover operation

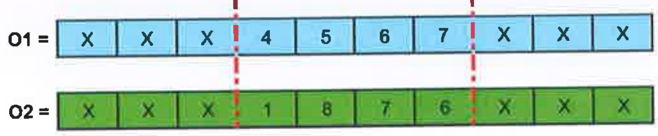
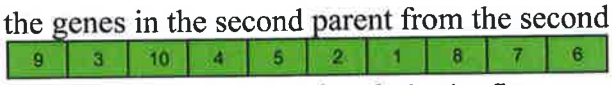
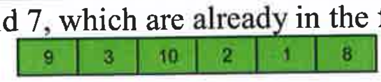


Figure B-2 copping the selected fragments into the corresponding offspring

- The sequence of the genes in the second parent from the second cut point is:



- Removing genes 4,5,6 and 7, which are already in the first parent results in:



- This sequence is placed in the first offspring, starting from the second cut point:

O1 =	2	1	8	4	5	6	7	9	3	10

- The second offspring is obtained in a similar way:

O2 =	3	4	5	1	8	7	6	9	10	2

This operator preserves the order, adjacency and absolute positions of genes of the chromosome and relative order of the remaining genes. This operator is also used in TSP and suggest that the order of the cities are important and not their respective positions.

- **Cycle Crossover (CX)**: This operator is presented by [B-4]. It builds offspring in a way that each gene comes from one of the parents. The starting element of parent 1 is inherited by a child. The element, which is in the same position in parent 2 is considered and its position is found in parent 1 and then the gene is inherited by a child to the same position. This cycle is continued until encountering the initial element from parent 1 in parent2. The reaming genes are filled from the other parent. This is more evident by looking at the following example.

P1 =	1	2	3	4	5	6	7	8	9	10
P2 =	4	1	2	8	7	6	9	3	5	10

Figure B-3 Chromosomes before CX Crossover

Two parents would produce the first offspring by taking the first city for the first parent:

O1 =	1	X	X	X	X	X	X	X	X	X
O2 =	4	X	X	X	X	X	X	X	X	X

Figure B-4 Taking the first genes from each parents

Since every gene in the offspring should be taken from one of its parents from the same position then the next city to be chosen for offspring 1 is 4 from parent 2. The location of this gene in the first parent is on the forth segment. This implies gene 8 from parent 2 just below the selected gene 4. This process continuous until the cycle is completed.

O1 =	1	2	3	4	X	X	X	8	X	X
O2 =	4	1	2	8	X	X	X	3	X	X

Figure B-5 Completed cycles

The remaining genes are filled from the other parent:

O1 =	1	2	3	4	7	6	9	8	5	10
O2 =	4	1	2	8	5	6	7	3	9	10

Figure B-6 The remaining genes are filled

As it is evident from the above example the CX preserves the absolute position of the elements in the parent sequence.

B.3 Recent methods used to enhance optimisation techniques

As introduced in chapter 3 (section 3.4.9) the optimisation methods namely simulation annealing, tabu search and genetic algorithm can perform better. The following presents further survey on how the introduced optimisation methods can be further enhanced.

- Cordeau et al. [B-5] represents a tabu search algorithm that has very few controlled parameters in contrary to existing tabu searches presented in the literature. They indicate that their algorithm outperforms other tabu searches used to address multi-depot vehicle routing problems;
- Renaud et al. [B-6] describes a new tabu search algorithm called FIND. This algorithm consists of three phases: **F**ast **I**mprovement, **IN**tensifications and **D**iversification. Their computational results on 23 benchmark instances suggested that their approach outperforms exiting heuristics for VRSP problems;
- Gendreau et al. [B-7] describes a TS, which iteratively inserts clusters of vertices in the current tour for selection and it removes a chain of vertices. This algorithm was used for un-directed selective TSP and resulted near optimal solution more frequently;
- Lndrieu et al. [B-8] presented a probabilistic tabu search. Their computation results suggest their algorithm often produces optimal solutions in a relatively short execution time;
- Gunnels et al. [B-9] compared GA and SA heuristics on the gene mapping problem. They found that the GA method always converges to a good solution more quickly than the SA method. The best solution produced by the GA method is always superior to the best solution produced by the SA method;
- Low et al. (2004) [B-10]: presented an algorithm, which combines the merit of both SA and TS to solve the flow shop-scheduling problem. They developed a mechanism that records the good solution's characteristic into SA to make the searching procedure more robust. They showed that the proposed SA procedure performs well with respect to both solution and efficiency;
- Nearchou (2004) [B-11]: proposed a new hybrid SA approach for the flow shop-scheduling problem. This approach combines the SA procedure with features borrowed from the GAs such as use of population of individuals and recombination of operators. They report that their approach resulted in shorter computation time and higher performance than that of other meta-heuristic approaches in this field;
- Leung et al. (2003) [B-12] applied a GA and a simulated annealing-genetic algorithm to a two-dimensional orthogonal packing problem. They found that in the long run, the mixed heuristic produces better results, while the GA produces only good results bur in a shorter time.

B.3.1 Hypothetical Pareto GA example:

The aim here is to illustrate how the Pareto Optimal GA could help here to address monthly transportation scheduling problem. The following schedules are generated hypothetically and they are not based on real data. Also for simplicity, the two main objectives are considered here to minimise both distance travelled and inventory cost.

1. **Initialisations:** A number of schedules addressing monthly transportation problems are randomly generated. These schedules are evaluated based on their impact on different objectives considered for this problem.

2.

3. Table B-8, illustrates these schedules based on the common objectives.

Table B-8 Hypothetical transportation schedules

Schedule No.#	Travelling distances	Inventory Levels	Resource Utilisation	Service level	Environmental Impact	Completion time	Safety	Union	Fleet size
1	591134	100	85	85	150	30	5	5	40
2	515962	120	80	87	100	28	5	5	38
3	471383	80	75	83	125	28	10	15	35
4	335943	53	68	84	65	26	15	20	40
5	317171	76	80	87	78	25	10	25	36
6	451083	30	85	86	95	28	8	20	35

4. **Dominance Check**: The current population is checked based on the first two objectives to see which individuals are non-dominated. Figure B-7, illustrates, six scenarios compared on the bases of travelled distance and Inventory cost. Scanning the graph shows that the best points are lower on the page and to the left. In particular, Scenarios E, D and F seem like good possible choices: even though none of the three points is best along both dimensions. There are trade-offs from one of these three scenarios to another; there is a gain along one dimension and loss along others. These three points are none dominated because there are no points better than these on all criteria. On the other hand scenarios A, B and C seem to be poor choices. It is the case that these scenarios are dominated by another point. Scenario C [471383, 80] is dominated by either of E, D or F. Also, both scenarios B and A are dominated by scenario C. Thus in this problem instead of obtaining a single answer, there is a set of answers that are not dominated by any others, the Pareto Optima set {E, D, F}.

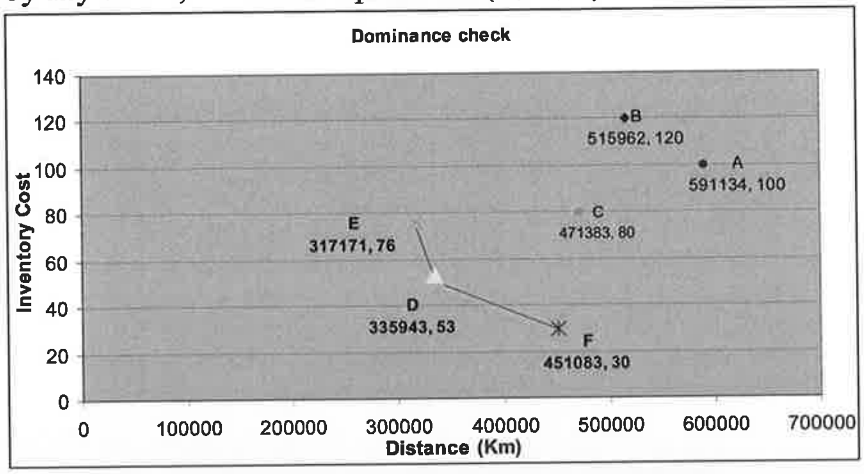


Figure B-7 illustration of Multi-objective optimisation

5. **Adjustment:** At this stage the dominated individuals {A, B, C} are replaced with offspring created from two different non-dominated parents. The parents are obtained randomly from the current non-dominated population with equal probability of selection. Figure B--8, shows the selected Pareto optimal set from the population and Figure B-9, illustrated the generated offspring based on random selection of parents and genetic operations.

Non-dominated	Distance	Inventory
D	335943	53
E	317171	76
F	451083	30

Figure B--8 Pareto Optima set

Parents	Offspring	Distance	Inventory
D,F	G	561484.5	68
D,E	H	485142.5	28
E,F	I	381000	40

Figure B-9 randomly generated offspring

6. **Dominance Recheck:** The generated solutions are rechecked to check for the new Pareto optimal set. As shown in Figure B-10, the Pareto optimal set is {D, I, F, H} and the only dominated solution is G, which is to be removed from the data set.

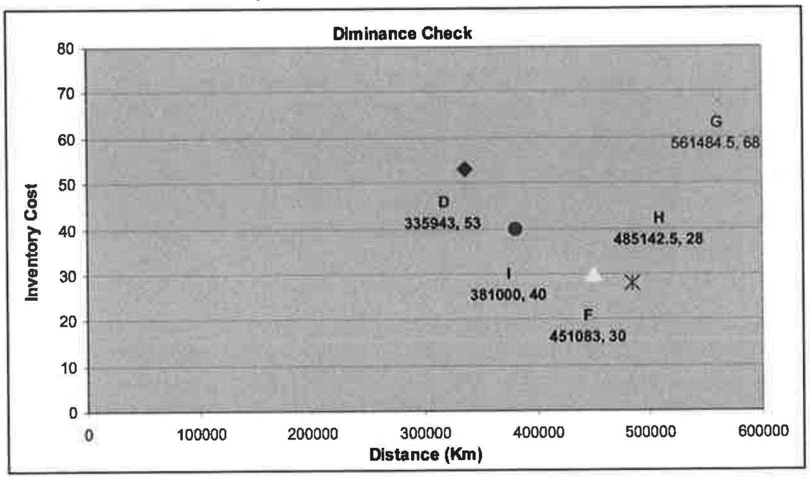


Figure B-10 Dominance check for Pareto Optimal set

5. **Increase population**: The Pareto optimal set contains 5 individuals. There is a need of one more non-dominated solution to be generated from the current Pareto population. Therefore, the adjustment stage is repeated and if it ends in a Pareto set of 6 individuals then another 6 new individuals are randomly generated and these are added to the current populations.
6. **Repeat:** The algorithm goes back to step 2 for further dominance checks.

B.4 Recent development in Genetic algorithm

As indicated in chapter 3 (section 3.7) considerable research has been conducted to improve the GA performance. The aim of this section is to further provide detail on the developed methods.

B.4.1 Selection Method:

The population diversity and selective pressure are two important issues in the evolution process of the GA search. These are highly related parameters in that a strong selective pressure supports the premature convergence of the search and weak selective pressures make the search inefficient. Much research has been conducted to introduce selection methods to strike a balance between these two factors [B-13, B-14, B-15, B-16].

To improve selection mechanism [B-15] in his early work introduced several variations of simple selection methods. These variations are: The *elicit model*, which enforces preserving the best chromosome. The *expected value model*, which is used to reduce the stochastic errors of the selection routine. The *elitist expected value model*, which combines the first two methods together and finally, the *crowding factor model* that allows a newly generated chromosome to replace the old one. The rejected chromosome is selected from those, which resemble the new one.

Also, [B-14] introduced a number of modifications in selection methods, which resulted in superiority of some of these modifications over the simple selection. These modifications are such as: *deterministic sampling, remainder stochastic sampling without replacement, stochastic tournament* and *remainder stochastic sampling with replacement.* The latter was proven to be the most successful one and it was adopted by many researchers. Later in 1987 [B-13] introduced *stochastic universal sampling* as a further modification of the sampling methods.

Also, other methods were introduced based on the belief that the common cause of rapid premature convergence is mainly due to the presence of highly fit chromosomes, which are much better than the average fitness of the population. Also, a method using artificial weight to sample a population and two other schemes that allow the user to influence the selective pressure parameter of the algorithm were presented in [B-1]. In addition tournament selection [B-16] was introduced which combines the idea of ranking in an interesting and efficient way. Some of the recent alternative methods introduced as selection methods in the literature are presented as follows:

- Chul et al. [B-17] proposed a new genetic evaluation and selection mechanism called a Pareto Stratum-niche cubicle. This selection method associates every individual with a rank. The rank is determined by the sparseness of individuals and Pareto Optimality.
- Sang-Keon Oh et al. [B-18], proposed a new selection method based on a non-linear fitness assignment. This method uses a ranking selection, which permits a higher local exploitation search, where the diversity of the population is maintained by parallel subpopulation structures.
- Wang et al. [B-19], introduced a new evolving mechanism to improve the solution quality and searching efficiency. This mechanism extracts a generalise pattern from elite chromosomes in the population. In this mechanism the difference between individual chromosomes and the pattern is calculated and used to determine selection probabilities of individuals. This probability is used to judge which chromosome survives or dies.
- Tokoro [B-20] proposed a new genetic algorithm, which uses Welch's test in its selection stage to limit the probability that good chromosomes are excluded by sampling error. This algorithm excludes the individuals that are statistically inferior to the best individual in the current generation. Therefore, the probability that the best individual generated in a GA search process is removed at any generation by sampling error is limited.
- Dukkipati et al. [B-21] purposed Cauchy criteria for choosing the Boltzmann selection scheme. This is based on the hypothesis that selection strength should increase as the evolutionary process advances and the distance between two-selection strengths should decrease for the process to converge.
- Also, Tobias and Lothar [B-22] used the fitness distribution of a population as a new mathematical formulation to analyse the properties of common selection methods such as tournament, ranking and truncation selections. They introduced new factors such as *selection intensity*, *selection variance* or the *loss of diversity* that gave new insights into the properties of the selection methods. They also suggested, that in addition to these

parameters there is a need for further information about the recombination phase of the algorithm to evaluate the behaviour of the selection methods.

B.4.2 Reproduction Methods:

The relative importance of crossover and mutation has long been studied in the literature of evolutionary computation. The traditional view is that crossover is primarily responsible for improvements in fitness and the mutation serves as a secondary role of introducing diversity into the population. Some researchers [B-23] presented theoretical arguments and empirical results to suggest that mutation can be more useful. In contrast some others argued in favour of crossover [B-24]. However, [B-25] indicates that the balance among crossover, mutation and selection is of great importance rather than a choice between crossover and mutation.

The proper choice of crossover and mutation operators is critical for the successful implementation of genetic algorithms. Different crossover and mutation operators are suitable for different problems even for different stages of the GA process in a problem. Determining which crossover and mutation operators should be used and also their respective rates are quite difficult tasks and they are usually performed by trial and error. [B-26] provided a number of guidelines for setting the crossover and mutation rates. There are several methods that could be used to adjust the ratios of the crossover operators. Also they used a varying mutation rate suggesting that a mutation rate decreased exponentially over generations had superior performance. To this effect [B-27] cyclically varied the mutation rate to demonstrate similar results.

In addition, [B-26] proposed a Dynamic Genetic algorithm (DGA) that simultaneously uses more than one crossover and mutation operator to generate the next generation. Also, the ratios of these operators are changed along with the evaluation results of the respective offspring in the next generation. They suggest that this algorithm performs better than the algorithms with single crossover and mutation operators.

Generally, reproduction operations involve recombination of two parents to produce offspring in a genetic algorithm. However in recent work attempts were made to study the effect of using more than two parents for recombination. The *global recombination in evolution strategy* (ES) [B-28] was one of the early works in this area that allowed generation of one new individual, which may inherit genes from more than two parents.

The *scanning crossover* and *diagonal crossover* were originally introduced as initial multi-parent recombination operators [B-29]. In addition, [B-28] proposed three types of multi parent recombination operators: the *centre mass crossover (CMC)*, *Multi parent feature-wise crossover (MFC)* and *Seed Crossover (SX)*. They suggest this types of recombination lead to better performance. Also, [B-23] introduced an Adaptive *Neighbourhood based Multi parent Crossover operator (ANMX),* which generates offspring using linear non-convex combination of several relative parents. This operator not only takes on the form of a multi-parent crossover operator, but also represents some characteristics of a mutation operator. Examples of other multi-parent crossovers are: *Blend crossover* [B-30], *Simplex crossover* [B-31] and *GUO's Crossover* [B-23]. The above-mentioned Crossover mechanisms were introduced as improvements on the single point crossover technique.

B.4.3 Encoding

The classical GAs are not appropriate tools for finding the optimal solutions with the desired precision. This could be due to the premature convergence to a non-global optimum, inability to perform fine local tuning or inability to operate in the presence of nontrivial constraints. In the last decade there have been some attempts to improve this drawback by addressing the parameter representation in the GA.

B.4.3.1 Gray Coding:

Any parameter optimisation technique, including GAs, requires some method of representing parameters such as integer, bit string and floating point representations. Gray coding is a type of bit string representation. The main objective of this coding method is to move the genetic algorithm closer to the problem space. This representation has the property that any two points next to each other in the problem space differ by one bit only. Therefore an increase of one step in the parameter value corresponds to a change of a single bit in the code. This is not generally achievable in the binary approach. Also, there have been some other attempts to improve parameter representations of the GA during the last decade as introduced below:

B.4.3.2 Delta Coding:

This modification of GA was proposed by [B-32]. In this method individuals in a population are not treated as a potential solutions but rather as additional small values (i.e. delta values), which are added to the current potential solution. This algorithm applies the GA techniques on two levels known as: the level of potential solutions to the problem (level χ) and the iterative phase, the level of delta changes (level δ). Figure B-11 illustrates this algorithm. Delta coding makes mutation unnecessary, due to re-initialisation of the populations on level δ for each iteration.

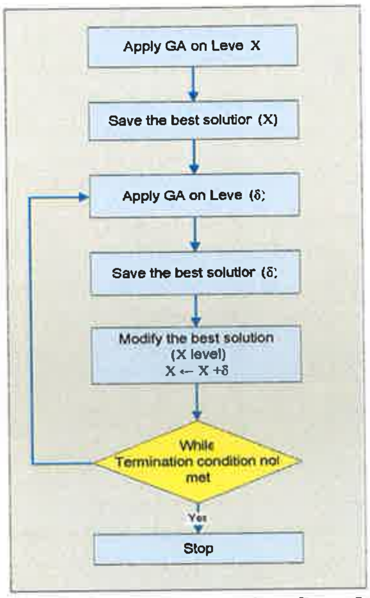


Figure B-11 A Delta coding flowchart

B.4.3.3 Dynamic Parameter Encoding:

This strategy was proposed by [B-33]. In this method, the accuracy of the encoded parameters are dynamically adjusted to increase the resolution of the solution and to focus on the most promising area of the search space. In this system each component of the solution vector is represented by a fixed-length binary string, which represents the precision of the individual. When in some iteration a genetic algorithm converges, the most significant bit of the solution is saved and then dropped. The remaining bits are shifted one position left, and a new bit is introduced. This bit increases precision by a finer partitioning of the search space. This process is repeated until some global condition is met.

B.4.3.4 ARGOT Strategy:

The ARGOT (Adaptive Representation Genetic Optimiser Technique) is designed to solve problems using a genetic algorithm. In this method binary strings are used to represent search parameters. Intermediate mapping is performed between the bit strings and the search space. This mapping is based on internal measurements such as parameter *convergence*, *variance* and *positioning* within a possible range of parameter values. These measurements are used to drive strategies within ARGOT such as *adjusting parameter resolution* and *drifting parameter boundary locations* [B-34].

B.4.3.5 MICRO-GAs:

Goldberg [B-35] first described this method for serial GA. This method considers a small population. The population is measured for convergence either by genotype or phenotype convergence. If the population has converged the best string is kept and the rest of the population is randomly regenerated and the search is repeated. Also, [B-35] states that Micro-GAs are like delta coding in that the population is reinitialised and prior results are included in the population by saving the best string. In addition one disadvantage of micro-GA is that the small population prevents adequate hyper plane sampling and simple hill climbing may dominate the search.

B.4.4 Adaptive Techniques:

Most of the evolutionary algorithms such as GA use fixed representation operators and control parameters. Most recent promising research areas are based on inclusion of self-adapting mechanism. These mechanisms are mainly used for [B-36]: chromosome representation, reproduction operators and control parameters.

[B-26] suggest that for many applications the crossover and mutation operators adapted are the key to the success of the GAs. [B-37], used an adaptive genetic algorithm in which P_c and P_m were varied according to the fitness values of the solutions. In contrary, [B-38] suggested adaptive crossover operations rather than varying their respective rates. Examples of these operators are: selective crossover, adaptive uniform crossover and Masked Crossover.

In addition [B-26] presented several methods for adjusting the ratios of the crossover operators. They also indicated a geometric progression method, which was used to adjust the ratios of the crossover operators in the geometric progression. [B-39] proposed a probabilistic rule-driven adaptive model (PRAM) for parameter adaptation and a repelling approach for diversity maintenance in GAs.

B.4.5 Multiple population

One of the most important concepts in GA is multiple population. Multi-population Ggenetic Algorithms (MGAs) are considered as an extension of traditional single population GAs (SGAs). SGA is powerful and performs well on many problems. However, introducing many populations (i.e. sub-populations) could help to obtain better results.

MGAs divide the population into several isolated sub-populations. In MGAs: first, each sub-population evolves independently for specific number of generations (i.e. isolation time) to reach the theoretical equilibrium obtaining many local optimal solutions in sub-populations. After the isolation time, each sub-population exchanges a number of individuals with its neighbourhoods, which is known as migration. The migration would interrupt the equilibrium and forces each sub-population to evolve again to escape the local optimum and therefore exploiting toward the global optimum. The number of exchanged individuals is known as migration rate and the scheme of migration determines how much genetic diversity can occur in the sub-populations and the exchange of information between sub-populations.

Researchers [B-40, B-41] have found MGAs are more effective both in speed and solution quality such as:

- They would shorten the number of generations needed to find the optimal or near optimal solution.
- They usually can find more that one optimal solution,
- They have more resistance to premature convergence.

However, the behaviour and performance of MGAs are still heavily affected by an appropriate choice of parameters such as [B-42, B-40]: connection topology, migration method, population number and migration interval. To this effect [B-43] developed self-adaptive MGAs to adjust optimal parameters.

In many problems addressed by GA, the fitness evaluation for each candidate solution could be calculated independently. This means that each candidate solution can be calculated at the same time or in parallel. Based on this concept, the Parallel Genetic algorithms (PGAs) were developed to speed up computation by using the power of parallel computers. Such approaches exploit the parallel nature of GA methods in order to split the overall computation task into several smaller tasks, each of which may be performed on an individual computer processing element. In general, a number of PGA implementations have been suggested such as [B-44]: Single-population master-slave, Single-population Fine-grained, Multiple-population coarse-grained. Several researchers have shown parallel/multi-population GA for different applications yield better performance than single population implementations [B-45, B-46].

References B

B - 1. Baker, J.E., **Adaptive selection methods for genetic algorithms.** In Proceeding of the first International conference on Genetic algorithms, (1985), PP: 101-11.

B - 2. Goldberg, D.E., ***Genetic algorithms in search, optimisation, and machine learning.*** Addison Wesley Longman, Inc, (1953), USA, ISBN: 0-201-15767-5.

B - 3. Davis, L., ***Handbook of genetic algorithms.*** Van Nostrand Reinhold, (1991), New York.

B - 4. Oliver, I.M., Smith, D.J., Holland, J.R.C., **A study of permutation crossover operators on the travelling salesman problem.** Genetic algorithms and their applications proceedings of the second International Conference on Genetic algorithms, (1987), PP: 224-30.

B - 5. Cordeau, J.F., Gendreau, M., Laporte, G., **A tabu search heuristic for periodic and multi-depot vehicle routing problems.** Journal of Networks, Vol. 30, Issue 2, (1997), PP: 105-19.

B - 6. Renaud, J., Laporte, G., Boctor, F.F., **A tabu search heuristic for the multi-depot vehicle routing problem.** Journal of Computers Operations Research, Vol. 23, Issue 3, (1996), PP: 229-35.

B - 7. Gendreau, M., Laporte, G., Semet, F., **A tabu search heuristic for the undirected selective travelling salesman problem.** European Journal of Operational Research, Vol. 106, (1998), PP: 539-45.

B - 8. Lndrieu, A., Mati,Y., Binder,Z., **A tabu search heuristic for the single vehicle pickup and delivery problem with time windows.** Journal of Intelligent Manufacturing, Vol. 12,(2001), PP:497-508.

B - 9. Gunnels, J., Cull, P., Holloway, J.L., **Genetic algorithms and simulated annealing for gene mapping.** Evolutionary Computation, IEEE World Congress on Computational Intelligence, Proceeding of the First IEEE conference, Vol. 1, (1994), PP: 385-90.

B - 10. Low, C., Yeh, J.J., Huang, K.I., **A robust simulated annealing heuristic for flow shop scheduling problems.** International Journal of Advance Manufacturing Technology, Vol. 23, (2004), PP: 762-7.

B - 11. Nearchou, A.C., **Flow-shop sequencing using hybrid simulated annealing.** Journal of Intelligent Manufacturing, Vol. 15, (2004), PP: 317-28.

B - 12. Leung, T.W., Chan, C.K., Troutt, M.D., **Discrete optimisation, application of a mixed simulated annealing-genetic algorithm heuristic for the two-dimensional packaging problem.** European Journal of Operational Research, Vol. 145,(2003), PP:503-42.

B - 13. Baker, J.E., **Reducing bias and efficiency in the selection algorithm.** Proceeding of the second International Conference On Genetic algorithms, (1987), PP: 14-21.

B - 14. Brindle A., **Genetic algorithms for function optimisation** (dissertation). USA: University of Alberta, Edmonton; 1981.

B - 15. De Jong KA. **An analysis of the behaviour of a class of genetic adaptive systems** (dissertation). University of Michigan, 1975.

B - 16. Goldberg, D.E., Deb, K., Korb, B., **Do not worry, be messy.** Proceeding of the fourth international conference on Genetic algorithms, (1991), PP: 24-30.

B - 17. Hyun, C.J., Kim, Y., Kim, Y.K., **A genetic algorithm for multiple objective sequencing problems in mixed assembly lines.** Computers Operations Research, Vol. 25, Issue 7/8, (1998), PP: 675-90.

B - 18. Oh, S.H., Kim, C.T., Lee, J.J., **Balancing the selection pressures and migration schemes in parallel genetic algorithms for planning multiple paths.** Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), Vol. 4, Issue 4, (2001), PP: 3314-9, ISBN: 0 7803 6576 3.

B - 19. WANG, Q., YUNG, K.L., IP, W.H., **A pattern-based evolving mechanism for genetic algorithm to solve combinatorial optimisation problems.** IEEE International Workshop on Soft Computing in Industrial Applications, Binghamton University, Binghamton, (2003), PP: 97-101, ISBN: 0-7803-7855-5.

B - 20. Tokoro, K.I., **A statistical selection mechanism of GA for stochastic programming problems.** Proceeding of the 2001 Congress on Evolutionary Computation (IEEE Cat.No.01TH8546), Vol. 1, Issue 1, (2001), PP: 478-92.

B - 21. Dukkipati, A., Murty, M.N., Bhatnagar, S., **Cauchy annealing schedule: An annealing schedule for Boltzman selection scheme in evolutionary algorithms.** Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), Vol. 1, Issue 1, (2004), PP: 55-62, ISBN: 0 7803 8515 2.

B - 22. Blickle, T., Thiele, L., **A comparison of selection schemes used in evolutionary algorithms.** Evolutionary Computation, Vol. 4, Issue 4, (1996), PP: 361-94.

B - 23. Li, J., Kang, L., Wu, Z., **An adaptive neighbourhood-based multi-parent crossover operator for real-coded genetic algorithms.** 2003 Congress on Evolutionary Computation (IEEE Cat. No.03TH8674), Vol. 1, Issue 1, (2003), PP: 14-21, ISBN: 0 7803 7804 0.

B - 24. Spears, W.M., ***Crossover or mutation? in foundations of genetic algorithms 2.*** L.D. Whitley. Morgan Kaufman, (1993).

B - 25. Mitchell, M. , **An introduction to genetic algorithms.** The MIT Press, (1996).

B - 26. Hong, T.P., Wang, H.S., Lin, W.Y., Lee, W.Y., **Evolution of appropriate crossover and mutation operators in a genetic process.** Applied Intelligence, Vol. 16, (2002), PP: 7-17.

B - 27. Hoen, T.P., Pettey, C.C., **Parental and cyclic-rate mutation in genetic algorithms: An initial investigation.** In Proceeding of Genetic and Evolutionary Computation Conference, (1999), PP: 297-304.

B - 28. Tsutsui, S., Ghosh, A., **A study on the effect of multi-parent recombination in real coded genetic algorithms.** Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, (1998), PP: 828-33.

B - 29. Eiben, A.E., Schippers, C.A., **Multi-parent's niche: N-array crossover on NK-landscapes.** Proceeding of the Parallel Problem Solving from Nature IV, (1996), PP: 319-28.

B - 30. Eshelman, L.J., Schaffer, J.D., **Real-coded genetic algorithms and interval schemata.** Foundation of Genetic algorithms 2, (1993), PP: 187-202.

B - 31. Tsutsui, S., Yamamura, M., Higuchi, T., **Multi-parent recombination with simplex crossover in real coded genetic algorithms.** Proc, GECCCO'99, (1999).

B - 32. Whitley, D., Gordon,V.S., Mathias, K. ,***Lamarckian evolution, the Baldwin effect and function optimisation.*** Parallel Problem Solving from Nature - PPSN III, Davidor, Y.; Schwefel, H.P.; Manner, R., (Eds.), (1994), PP: 6-15, http://citeseer.ist.psu.edu/context/385373/525278.

B - 33. Schraudoph, N., Belew, R., **Dynamic parameter encoding for genetic algorithms, machine learning.** Vol. 9, Issue 1, (1992), PP: 9-21, ISBN: 0885-6125.

B - 34. Shaefer, C., **The ARGOT strategy: Adaptive representative genetic optimiser technique.** Proceeding of the second International conference in Genetic algorithms and their applications, (1987).

B - 35. Whitley, D., Mathias, K., Fitzhorn, P., **Delta coding: an iterative search strategy for genetic algorithms.** Proceedings of the International Conference on Genetic algorithms, (1991), PP: 77.

B - 36. Michalewicz, Z., ***Genetic algorithms + data structures = evolution programs.*** 3rd. ed. Springer-Verlag, (1996), Berlin, New York, ISBN: 3540606769.

B - 37. Srinivas, M., Patnaik, L.M., **Adaptive probabilities of crossover and mutation in genetic algorithms.** IEEE Transactions on System, Man, and Cybernetics, Vol. 24, Issue 2, (1994), PP: 17-26.

B - 38. Vekaria, K., Clark, C., **Biases introduced by adaptive recombination operators.** In Proceedings of Genetic and Evolutionary Computation Conference, (1999), PP: 670-7.

B - 39. Wong, Y.Y., Lee, K.H., Leung, K.S., Ho, C.W., **A novel approach in parameter adaptation and diversity maintenance for genetic algorithms.** International Journal of Soft Computing, Vol. 7, Issue 1, (2003), PP: 506-15.

B - 40. Lin, W.Y., Lee, W.Y., Pei Hong, T., **On self-adaptive multi-population genetic algorithms.** IEEE International Conference on Systems, Man and Cybernetics. Conference Proceedings (Cat. No.02CH37349), Vol. 6, Issue 6, (2002), PP: 6, ISBN: 0 7803 7437 1.

B - 41. The multipopulation genetic algorithm: Local selection and migration [homepage on the Internet]. Available from: http://www.pohlheim.com/Papers/mpga_gal95/gal2_1.html.

B - 42. Cantu-Paz, E., Goldberg, D.E., **Efficient parallel genetic algorithms: Theory and practice.** Computer Methods in applied mechanics and engineering, Vol. 186,(2000), PP:221-38.

B - 43. Yang, W., Lee,W.Y., Hong,T.P. ,**On self-adaptive multi-population genetic algorithms.** 2002 IEEE International Conference on Systems, Man and Cybernetics. Conference Proceedings (Cat. No.02CH37349), Vol. 6, Issue 6, (2002), PP: 6, ISBN: 0 7803 7437 1.

B - 44. Habershon, S., Harris, K.D.M., Johnston, R.L., **Development of a multi-population parallel genetic algorithm for structure solution from powder diffraction data.** Journal of Computational Chemistry, Vol. 24, Issue 14, (2003), PP: 1766-74, ISBN: 0192-8651.

B - 45. Kazerony, M., Lohmann, R., **Application of evolution strategy in parallel populations.** Proceedings of Parallel Problem Solving from nature, (1991), PP: 198-208.

B - 46. Voigt, H.M., Santibanez-koerf,I., Born, J. ,**Hierarchically structured distributed genetic algorithm .** Proceeding of Parallel Problem Solving from Nature 2, (1992), PP: 145-54.

Appendix C: GA Design and Developments

C.1 The Search Engine User interface Design

The followings describe the main features of the search engine user interface design.

C.1.1 System Descriptions

The main purpose of this menu option is to define any real or hypothetical transportation systems. In this approach these systems can be defined in terms of supply sources, demand centres and transportation resources such as trucks. The user can specify the sources and demand nods and different possible routes or links connecting these nodes. Also, transportation resources can be specified here. Further to defining the main components of the system, one can specify the available inventory, inventory shortage and holding costs and location of supply and demand nodes. In addition, particular specifications for routs such as travelling distance and expected travel time, fatigue, safety and environmental cost factors can be specified here. In the case of transportation resources, one could specify two types of fleets either owned or contracted ones. For each truck types, the fixed and operational costs could be specified. Figure C-12, illustrates the menu options that user can access to describe a system.

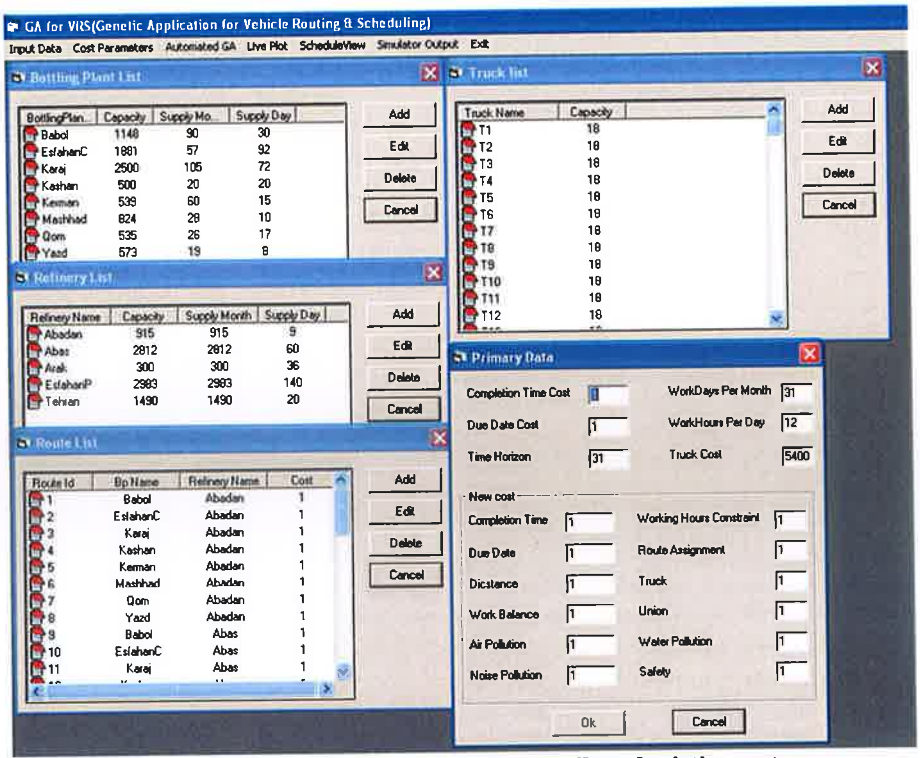


Figure C-12 Menu options used to describe a logistics system

C.1.2 Live Plot

This option allows one to monitor the GA performance as it proceeds along the search space. The general statistics from each proceeding generations are plotted against the simulation time. These statistics are such as minimum, maximum, average and sum of the fitness in a generation. Also fitness variance and standard deviations could be plotted. Figure C-13, demonstrates the menu options to plot general statistics. The user has the option to choose any combination of these metrics for dynamic plots.

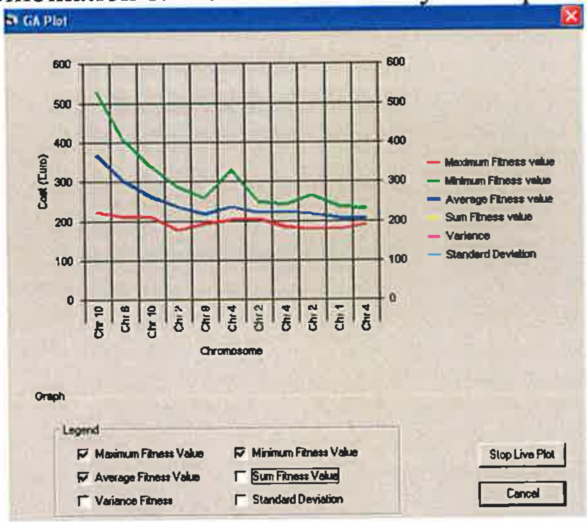


Figure C-13 Live Plot menu options

C.1.3 Schedule View

This menu option allows one to view the detail of the best chromosome generated during the simulation run. One can observe the generate transportation schedule for each considering truck. Also, total schedule cost and its breakdown are provided.

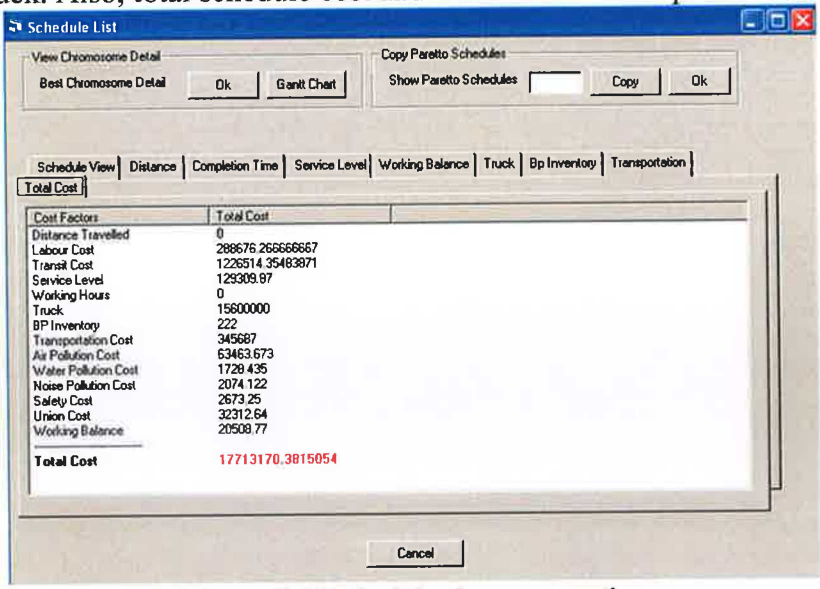
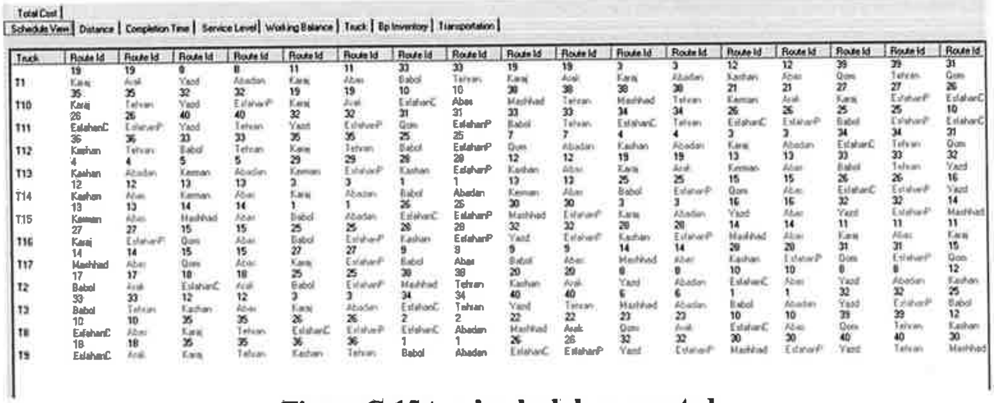


Figure C-14 Schedule view menu option



Truck	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id	Route Id
T1	19 Karaj	19 Arak	8 Yazd	8 Abadan	11 Karaj	11 Abas	33 Babol	33 Tehran	19 Karaj	19 Arak	3 Karaj	3 Abadan	12 Kashan	12 Abas	39 Qom	39 Tehran	31 Qom
T10	35 Karaj	35 Tehran	32 Yazd	32 EsfahanP	19 Karaj	19 Arak	10 EsfahanC	10 Abas	38 Mashhad	38 Tehran	38 Mashhad	38 Tehran	21 Kerman	21 Arak	27 Karaj	27 EsfahanP	26 EsfahanC
T11	26 EsfahanC	26 EsfahanP	40 Yazd	40 Tehran	32 Yazd	32 EsfahanP	31 Qom	31 EsfahanP	33 Babol	33 Tehran	34 EsfahanC	34 Tehran	26 EsfahanC	25 EsfahanP	25 Babol	25 EsfahanP	10 EsfahanC
T12	36 Kashan	36 Tehran	33 Babol	33 Tehran	35 Karaj	35 Tehran	25 Babol	25 EsfahanP	7 Qom	7 Abadan	4 Kashan	4 Abadan	3 Karaj	3 Abadan	34 EsfahanC	34 Tehran	31 Qom
T13	4 Kashan	4 Abadan	5 Kerman	5 Abadan	29 Kerman	29 EsfahanP	28 Kashan	29 EsfahanP	12 Kashan	12 Abas	19 Karaj	19 Arak	13 Kerman	13 Abas	33 Babol	33 Tehran	32 Yazd
T14	12 Kashan	12 Abas	13 Kerman	13 Abas	3 Karaj	3 Abadan	1 Babol	1 Abadan	13 Kerman	13 Abas	25 Babol	25 EsfahanP	15 Qom	15 Abas	26 EsfahanC	26 EsfahanP	16 Yazd
T15	13 Kerman	13 Abas	14 Mashhad	14 Abas	1 Babol	1 Abadan	26 EsfahanC	26 EsfahanP	30 Mashhad	30 EsfahanP	3 Karaj	3 Abadan	16 Yazd	16 Abas	32 Yazd	32 EsfahanP	14 Mashhad
T16	27 Karaj	27 EsfahanP	15 Qom	15 Abas	25 Babol	25 EsfahanP	28 Kashan	28 EsfahanP	32 Yazd	32 EsfahanP	28 Kashan	28 EsfahanP	14 Mashhad	14 Abas	11 Karaj	11 Abas	11 Karaj
T17	14 Mashhad	14 Abas	15 Qom	15 Abas	27 Karaj	27 EsfahanP	9 Babol	9 Abas	9 Babol	9 Abas	14 Mashhad	14 Abas	28 Kashan	20 EsfahanP	31 Qom	31 EsfahanP	15 Qom
T2	17 Babol	17 Arak	18 EsfahanC	18 Arak	25 Babol	25 EsfahanP	38 Mashhad	38 Tehran	20 Kashan	20 Arak	8 Yazd	8 Abadan	10 EsfahanC	10 Abas	8 Yazd	8 Abadan	12 Kashan
T3	33 Babol	33 Tehran	12 Kashan	12 Abas	3 Karaj	3 Abadan	34 EsfahanC	34 Tehran	40 Yazd	40 Tehran	6 Mashhad	6 Abadan	1 Babol	1 Abadan	32 Yazd	32 EsfahanP	25 Babol
T8	10 EsfahanC	10 Abas	35 Karaj	35 Tehran	26 EsfahanC	26 EsfahanP	2 EsfahanC	2 Abadan	22 Mashhad	22 Arak	23 Qom	23 Arak	10 EsfahanC	10 Abas	39 Qom	39 Tehran	12 Kashan
T9	18 EsfahanC	18 Arak	35 Karaj	35 Tehran	36 Kashan	36 Tehran	1 Babol	1 Abadan	26 EsfahanC	26 EsfahanP	32 Yazd	32 EsfahanP	30 Mashhad	30 EsfahanP	40 Yazd	40 Tehran	30 Mashhad

Figure C-15 truck schedule generated

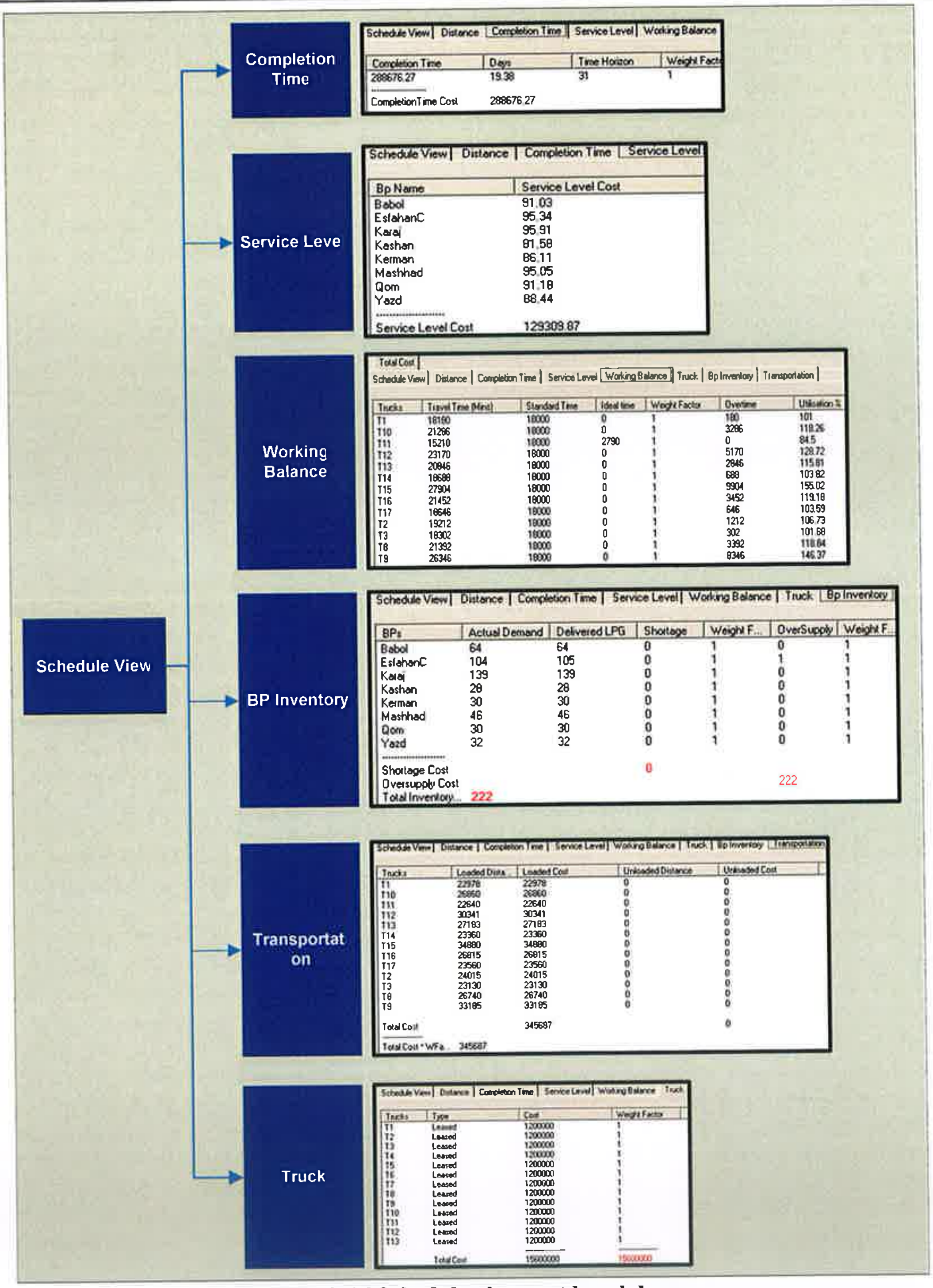


Figure C-16 Schedule view cost breakdowns

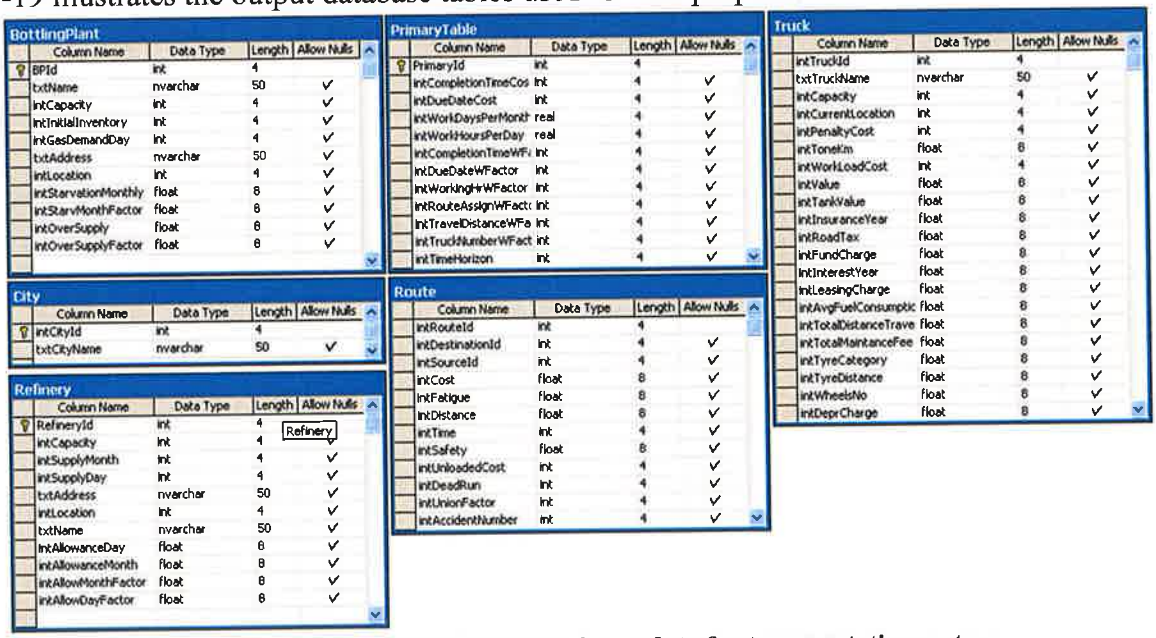
C.2 The Database-GA application Interface

The search engine works intensively with database to manage the storage and manipulation of data during the evolution to optimum solution. The database is composed of a series of related tables. These tables can be categorised as Input, processing and output data tables.

The input tables as shown Figure C-17 are used to store data from the user about the considering components of the system such as supply sources, demand centres, transportation fleet, routes and other primary parameters described earlier.

Also, when GA operations are taking place, there is a need of managing data processing and calculation as needed by different GA operations such as crossover, mutation, selection, elitism and others. Data held by chromosomes are preserved and updated in different tables such as ***RndEvaluation*** and ***Generation*** table. These processing database tables are illustrated in Figure C-18.

Finally, there are tables to record statistics and GA performance measures during the GA runs. The output tables contains information such as Generation statistics, optimum chromosomes, the Minimum/Maximum fit chromosomes per generations and etc. These tables could be used by the other applications to further analyse GA performance. Figure C-19 illustrates the output database tables used for this purpose.



BottlingPlant

Column Name	Data Type	Length	Allow Nulls
BPId	int	4	
txtName	nvarchar	50	✓
intCapacity	int	4	✓
intInitialInventory	int	4	✓
intGasDemandDay	int	4	✓
txtAddress	nvarchar	50	✓
intLocation	int	4	✓
intStarvationMonthly	float	8	✓
intStarvMonthFactor	float	8	✓
intOverSupply	float	8	✓
intOverSupplyFactor	float	8	✓

City

Column Name	Data Type	Length	Allow Nulls
intCityId	int	4	
txtCityName	nvarchar	50	✓

Refinery

Column Name	Data Type	Length	Allow Nulls
RefineryId	int	4	Refinery
intCapacity	int	4	✓
intSupplyMonth	int	4	✓
intSupplyDay	int	4	✓
txtAddress	nvarchar	50	✓
intLocation	int	4	✓
txtName	nvarchar	50	✓
intAllowanceDay	float	8	✓
intAllowanceMonth	float	8	✓
intAllowMonthFactor	float	8	✓
intAllowDayFactor	float	8	✓

PrimaryTable

Column Name	Data Type	Length	Allow Nulls
PrimaryId	int	4	
intCompletionTimeCos	int	4	✓
intDueDateCost	int	4	✓
intWorkDaysPerMonth	real	4	✓
intWorkHoursPerDay	real	4	✓
intCompletionTimeWF	int	4	✓
intDueDateWFactor	int	4	✓
intWorkingHrWFactor	int	4	✓
intRouteAssignWFact	int	4	✓
intTravelDistanceWFa	int	4	✓
intTruckNumberWFact	int	4	✓
intTimeHorizon	int	4	✓

Route

Column Name	Data Type	Length	Allow Nulls
intRouteId	int	4	
intDestinationId	int	4	✓
intSourceId	int	4	✓
intCost	float	8	✓
intFatigue	float	8	✓
intDistance	float	8	✓
intTime	int	4	✓
intSafety	float	8	✓
intUnloadedCost	int	4	✓
intDeadRun	int	4	✓
intUnionFactor	int	4	✓
intAccidentNumber	int	4	✓

Truck

Column Name	Data Type	Length	Allow Nulls
intTruckId	int	4	
txtTruckName	nvarchar	50	✓
intCapacity	int	4	✓
intCurrentLocation	int	4	✓
intPenaltyCost	int	4	✓
intTonekm	float	8	✓
intWorkLoadCost	int	4	✓
intValue	float	8	✓
intTankValue	float	8	✓
intInsuranceYear	float	8	✓
intRoadTax	float	8	✓
intFundCharge	float	8	✓
intInterestYear	float	8	✓
intLeasingCharge	float	8	✓
intAvgFuelConsumptic	float	8	✓
intTotalDistanceTrave	float	8	✓
intTotalMaintanceFee	float	8	✓
intTyreCategory	float	8	✓
intTyreDistance	float	8	✓
intWheelsNo	float	8	✓
intDeprCharge	float	8	✓

Figure C-17 Database tables to hold input data for transportation system

Figure C-18 Database Tables used to process GA operations

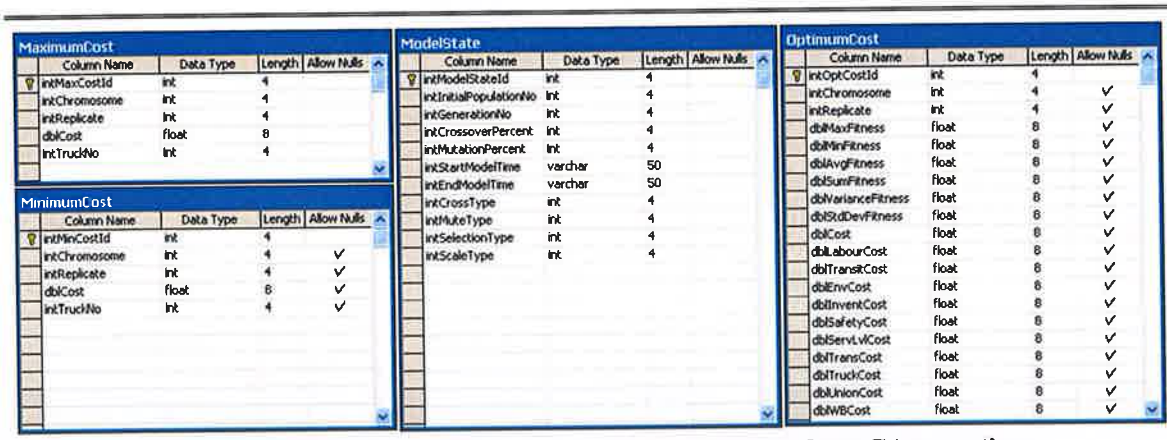


Figure C-19 Database Tables used to hold output data from GA operations

The Visual Basic program written here is based on class structure. Usually classes are used to encapsulate activities for different entities or objects considered in the program. Classes are also used to manage scatter data for different objects. There are two type of classes here, the business logic and data (i.e. properties) classes. Business logic classes are used to fetch data from SQL server into the VB program. Classes of these types are used to insert, delete or update activities during simulation run. For example getting a list of refineries currently available to transfer LPG from. In the other hand, Data classes are used to collect properties related to an entity considered in a program. For instance the initial data for refineries such as original capacity, location, monthly demands and etc. are collected using ***ClsRefineryData*** as shown in Figure C-21.

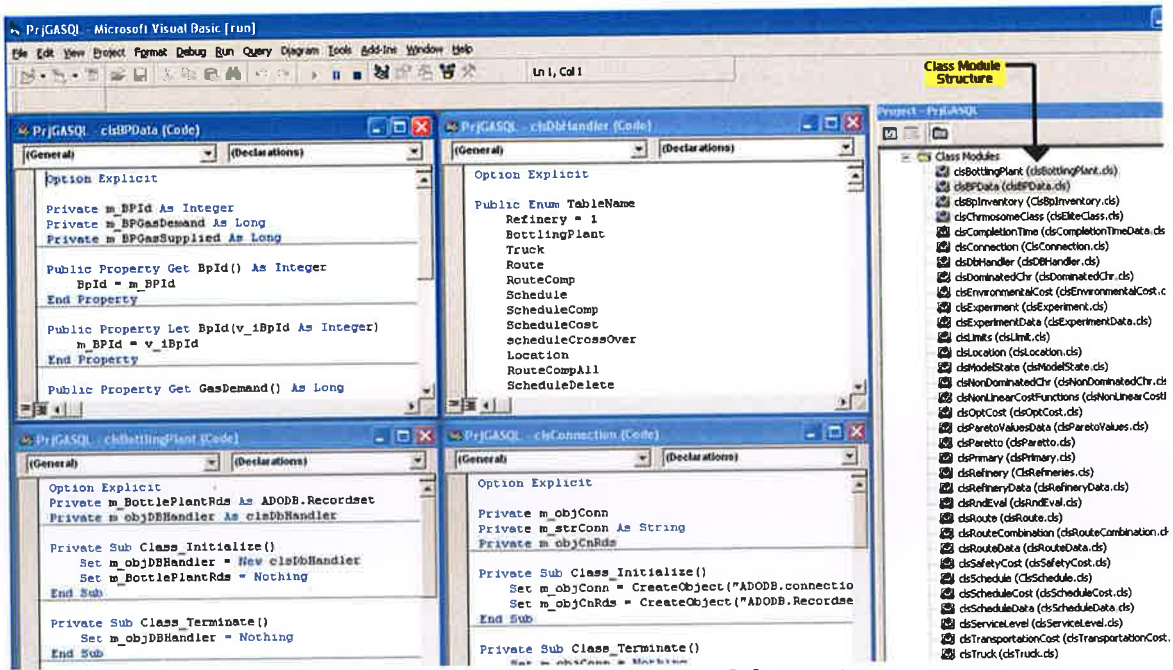


Figure C-20VB program Class module structure

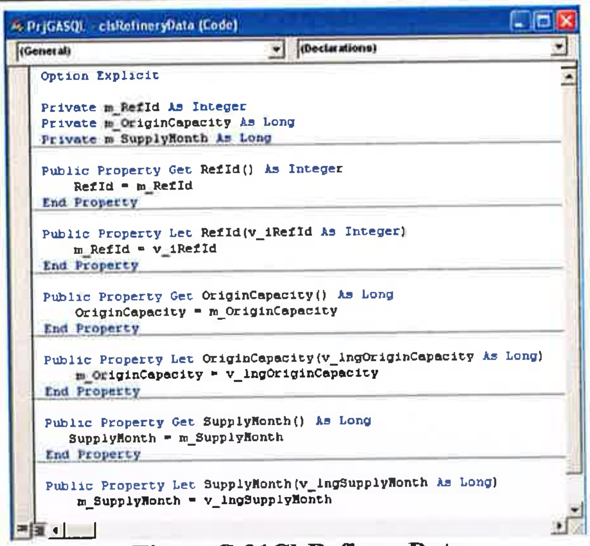


Figure C-21ClsRefineryData

Two major classes related to database are such as ***ClsConnection*** and ***ClsDBHandeller*** modules in this program. ***ClsConnection*** acts like a provider. This class is used to sets the database providers and allows connection to either MS Access database or MS SQL server as shown in Figure C-22. This class module allows record sets to be connected to Database for collecting data. Record sets are interfaces for collecting input data. The initial aim of ***ClsDBHandeller*** is to establish and generate record sets. For every record set there is a need of SQL command, which is done by this class module. Figure C-23 presents ***ClsDBHandeller*** interface and section C4 reviews the SQL commands provided in this module. Figure C-24 illustrates an example to further show how classes could be used to individual data processing.

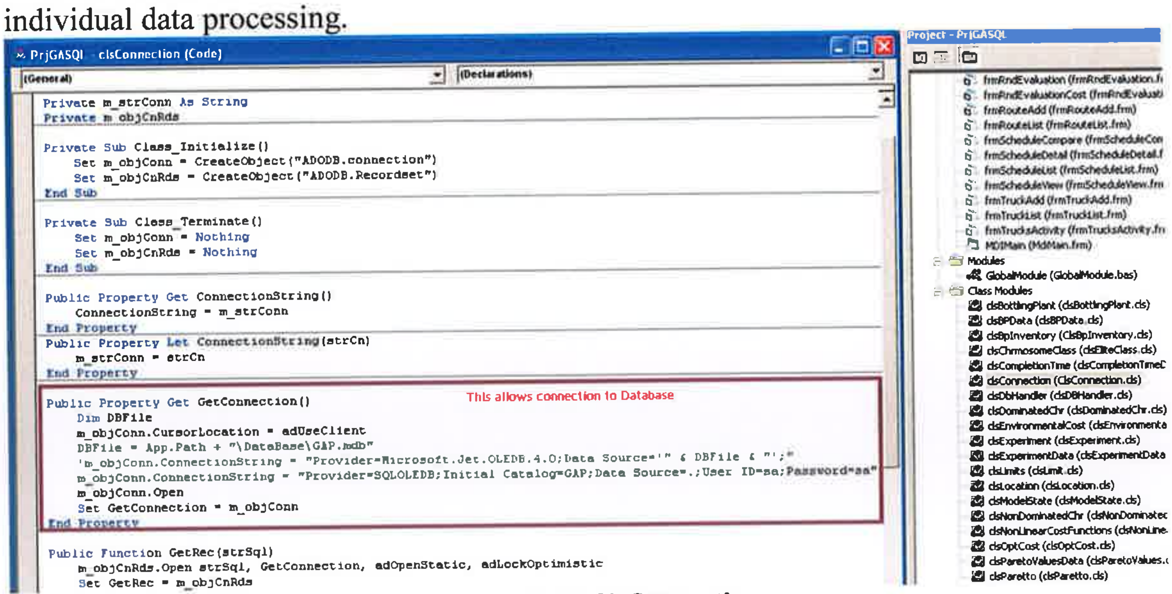


Figure C-22 ClsConnection

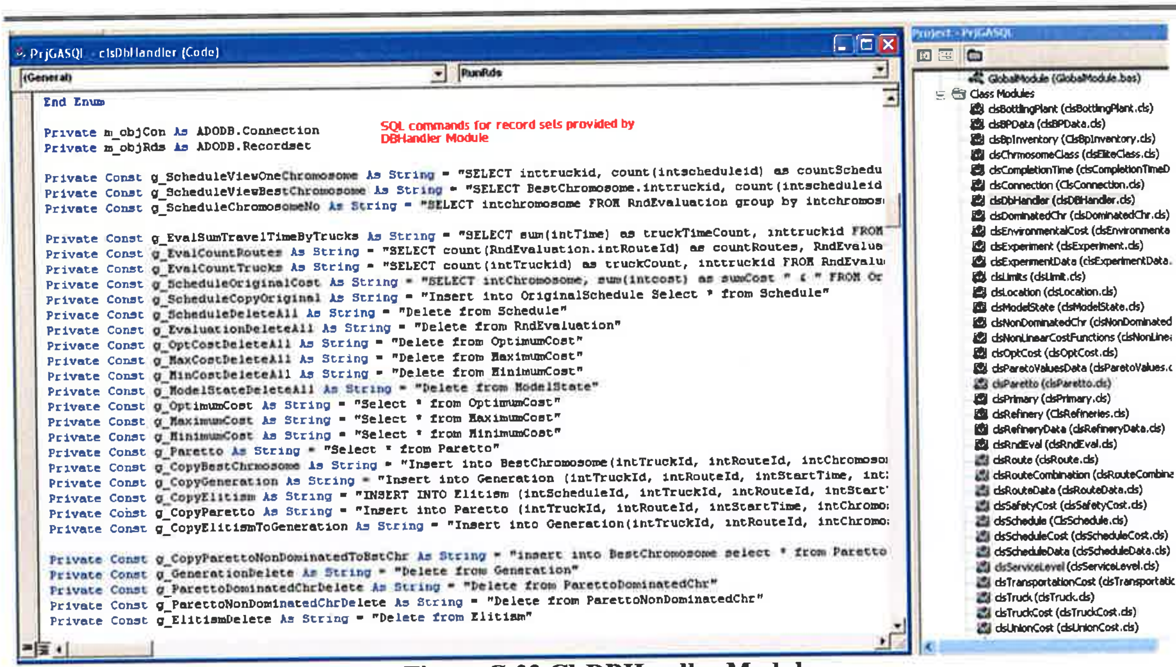


Figure C-23 ClsDBHandler Module

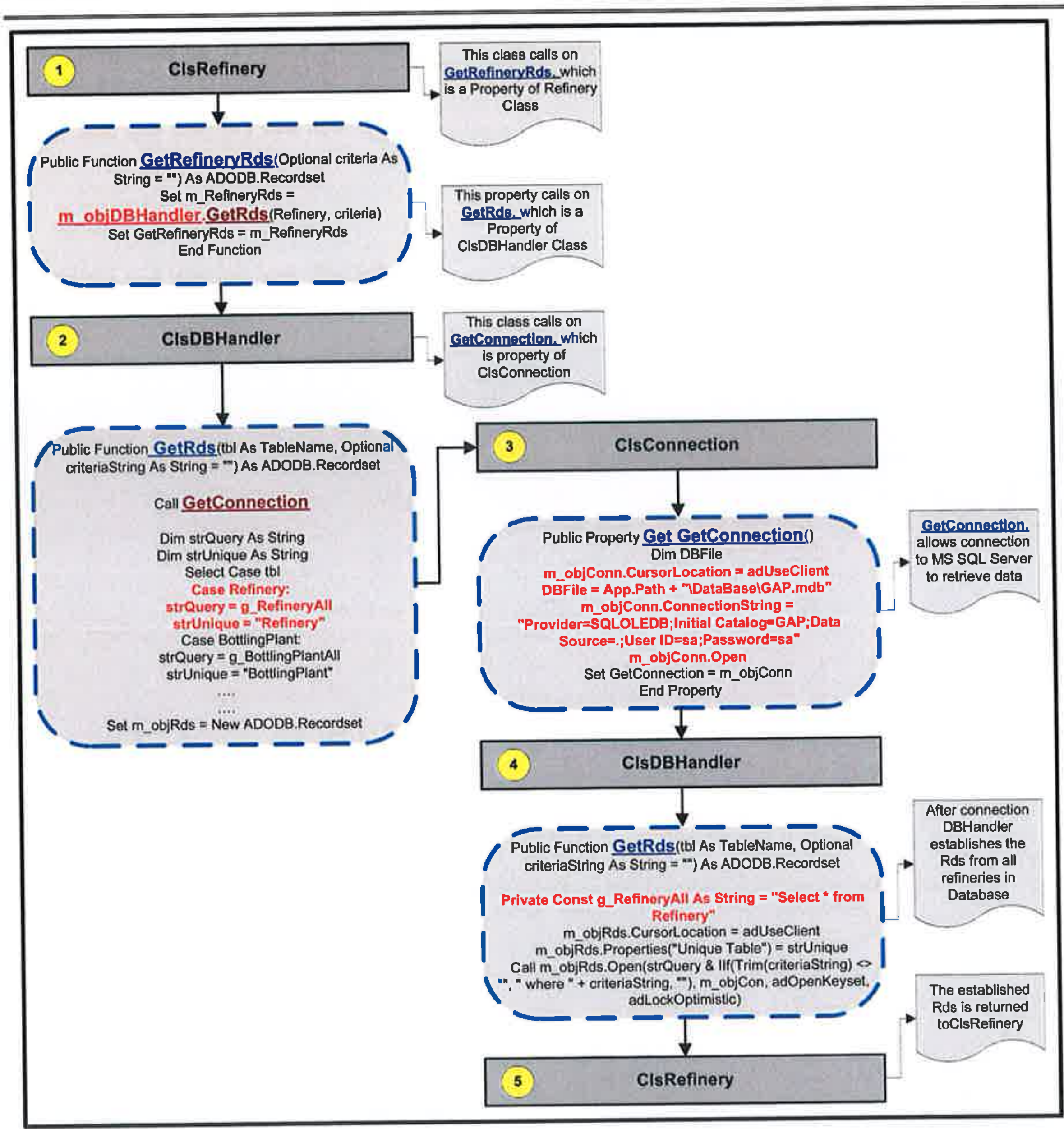


Figure C-24 Data Processing using classes

C.3 The Simulation tool -Database Interface

There are two links here. The first one is the link between GA to simulator and the second one the simulator to the GA as indicated in Figure C-25. These links are managed through database and the designed interfaces. The first link is to transfer optimum transportation schedule from the search engine to the simulator. The search engine application allows generated schedules from any of the methods to be selected and stored in a database for latter analysis or retrieval by the Witness simulation tool. This is mainly performed through the user interface designed as shown in Figure C-26.

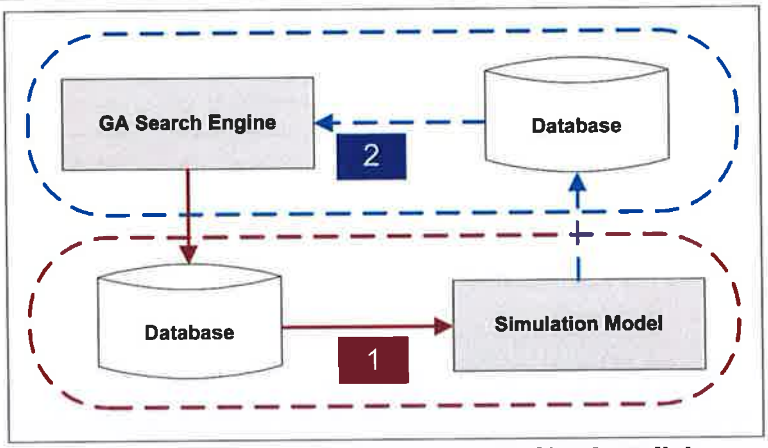


Figure C-25 Database Interface for GA-Simulator link

As illustrated in Figure C-26, the simulator output menu option allows the transfer of the data from the search engine to the simulator or any other applications. These data can be based on resources such as trucks, refineries or bottling plant used in obtained optimum schedule. Also, the ***BestChromosome*** table in the Database hold the detail of the optimum schedule. The content of this table is typically used and read in by the simulator to initiate the simulation run.

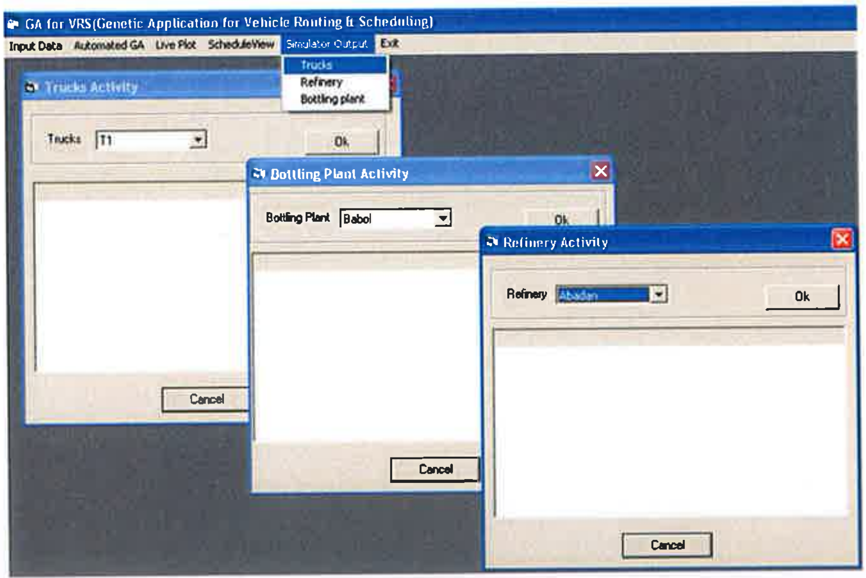


Figure C-26 Simulation outputs menu options

Data in Table 'BestChromosome' in 'GAP1' on '(LOCAL)'

intScheduleId	intTruckId	intRouteId	intChromosome	intStartTime	intStartTime1	intStartTime2	intTotalStartTime	intRank
1121674	28	37	7	880	0	0	880	0
1121675	35	35	7	52	0	0	52	0
1121676	30	23	7	120	0	0	120	0
1121677	24	4	7	100	0	0	100	0
1121678	31	7	7	960	0	0	960	0
1121679	21	18	7	240	0	0	240	0
1121680	19	18	7	240	0	0	240	0
1121681	22	11	7	1104	0	0	1104	0
1121682	14	33	7	256	0	0	256	0
1121683	29	14	7	1400	0	0	1400	0
1121684	13	11	7	1104	0	0	1104	0
1121685	20	18	7	240	0	0	240	0
1121686	18	1	7	1040	0	0	1040	0
1121687	23	11	7	1104	0	0	1104	0
1121688	27	21	7	720	0	0	720	0
1121689	17	1	7	1040	0	0	1040	0
1121690	26	12	7	1000	0	0	1000	0
1121691	32	39	7	104	0	0	104	0
1121692	16	17	7	544	0	0	544	0
1121693	15	9	7	1360	0	0	1360	0
1121694	25	4	7	100	0	0	100	0
1121695	34	38	7	720	0	0	720	0
1121696	15	3	7	800	0	0	2160	0
1121697	17	35	7	52	0	0	1092	0
1121698	30	35	7	52	0	0	172	0
1121699	24	29	7	562	0	0	662	0
1121700	29	40	7	680	0	0	2080	0
1121701	14	7	7	960	0	0	1216	0
1121702	35	18	7	240	0	0	292	0

Figure C-27 BestChromosome database table

However the second link is based on the simulation analysis. Simulation tool is used to evaluate the performance of the GA generated schedule. Based on the evaluation the initial parameters or the constraints used in GA runs are updated and then GA search is to be initiated to search for better optimum schedule incorporating changes from the simulator.

Currently, an expert does the evaluation and inferences from the simulation model. This task could be further automated using expert or adaptive systems. The simulation model highlights a number of short falls within the transportation schedule. Based on these inadequacies, constraints on GA are updated. This involves updating lower and upper limits that are re-set to initiate the GA search for better optimum schedules.

C.4 SQL Commands for GA-Database communication

1. Private Const g_ScheduleViewOneChromosome As String = ***"SELECT inttruckid, count[intscheduleid] as countSchedule FROM Schedule "***
2. Private Const g_ScheduleViewBestChromosome As String = ***"SELECT BestChromosome.inttruckid, count[intscheduleid] as countSchedule, txtTruckName FROM BestChromosome inner join truck on truck.intTruckId = BestChromosome.intTruckId"***
3. Private Const g_ScheduleChromosomeNo As String = ***"SELECT intchromosome FROM RndEvaluation group by intchromosome"***
4. Private Const g_EvalSumTravelTimeByTrucks As String = ***"SELECT sum[intTime] as truckTimeCount, inttruckid FROM RndEvaluation inner join Route on Route.intRouteId = RndEvaluation.intRouteId "***
5. Private Const g_EvalCountRoutes As String = ***"SELECT count[RndEvaluation.intRouteId] as countRoutes, RndEvaluation.intRouteId, intUpperLimit FROM RndEvaluation inner join Route on Route.intRouteId = RndEvaluation.intRouteId "***
6. Private Const g_EvalCountTrucks As String = ***"SELECT count[intTruckid] as truckCount, inttruckid FROM RndEvaluation "***
7. Private Const g_ScheduleOriginalCost As String = ***"SELECT intChromosome, sum[intcost] as sumCost " & " FROM OriginalSchedule INNER JOIN route ON route.intRouteId=OriginalSchedule.intRouteId where inttruckId <> 0 group by intChromosome"***
8. Private Const g_ScheduleCopyOriginal As String = ***"Insert into OriginalSchedule Select * from Schedule"***
9. Private Const g_ScheduleDeleteAll As String = ***"Delete from Schedule"***
10. Private Const g_EvaluationDeleteAll As String = ***"Delete from RndEvaluation"***
11. Private Const g_OptCostDeleteAll As String = ***"Delete from OptimumCost"***
12. Private Const g_MaxCostDeleteAll As String = ***"Delete from MaximumCost"***
13. Private Const g_MinCostDeleteAll As String = ***"Delete from MinimumCost"***
14. Private Const g_ModelStateDeleteAll As String = ***"Delete from ModelState"***
15. Private Const g_OptimumCost As String = ***"Select * from OptimumCost"***

16. Private Const g_MaximumCost As String = ***"Select * from MaximumCost"***
17. Private Const g_MinimumCost As String = ***"Select * from MinimumCost"***
18. Private Const g_Paretto As String = ***"Select * from Paretto"***
19. Private Const g_CopyBestChrmosome As String = ***"Insert into BestChromosome[intTruckId, intRouteId, intChromosome, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank] Select intTruckId, intRouteId, intChromosome, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank from RndEvaluation "***
20. Private Const g_CopyGeneration As String = ***"Insert into Generation [intTruckId, intRouteId, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank, intChromosome] Select intTruckId, intRouteId, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank, "***
21. Private Const g_CopyElitism As String = ***"INSERT INTO Elitism [intScheduleId, intTruckId, intRouteId, intStartTime, intChromosome]" & " SELECT RndEvaluation.intScheduleId, RndEvaluation.intTruckId, RndEvaluation.intRouteId, RndEvaluation.intStartTime, "***
22. Private Const g_CopyParetto As String = ***"Insert into Paretto [intTruckId, intRouteId, intStartTime, intChromosome, intStartTime1] Select intTruckId, intRouteId, intStartTime, intChromosome, intStartTime1 from RndEvaluation"***
23. Private Const g_CopyElitismToGeneration As String = ***"Insert into Generation[intTruckId, intRouteId, intChromosome, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank] Select intTruckId, intRouteId, intChromosome, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank from Elitism"***
24. Private Const g_CopyParettoNonDominatedToBstChr As String = ***"insert into BestChromosome select * from Paretto "***
25. Private Const g_GenerationDelete As String = ***"Delete from Generation"***
26. Private Const g_ParettoDominatedChrDelete As String = ***"Delete from ParettoDominatedChr"***
27. Private Const g_ParettoNonDominatedChrDelete As String = ***"Delete from ParettoNonDominatedChr"***
28. Private Const g_ElitismDelete As String = ***"Delete from Elitism"***
29. Private Const g_CopyGenerationToEvaluation As String = ***"Insert into RndEvaluation [intTruckId, intRouteId, intChromosome, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank] Select intTruckId, intRouteId, intChromosome, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank from Generation"***
30. Private Const g_TransferElitismToGeneration As String = ***"Insert into Generation[intTruckId, intRouteId, intChromosome, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank] Select intTruckId, intRouteId, intChromosome, intStartTime, intStartTime1, intStartTime2, intTotalStartTime, intRank from Elitism"***
31. Private Const g_ExperimentDelete As String = ***"Delete from Experiment"***
32. Private Const g_BestChrmosomeDeleteAll As String = ***"Delete from BestChromosome"***
33. Private Const g_ParettoScheduleDelete As String = ***"Delete from Paretto"***
34. Private Const g_OriginalScheduleDeleteAll As String = ***"Delete from OriginalSchedule"***
35. Private Const g_PrimaryAll As String = ***"Select * from PrimaryTable"***
36. Private Const g_ModelState As String = ***"Select * from ModelState"***
37. Private Const g_LocationAll As String = ***"Select * from City"***
38. Private Const g_RouteAll As String = ***"Select * from Route"***
39. Private Const g_RefineryAll As String = ***"Select * from Refinery"***
40. Private Const g_ScheduleAll As String = ***"Select intChromosome, intTruckId, intScheduleId, intRouteId from Schedule"***
41. Private Const g_ScheduleCrossOver As String = ***"Select intChromosome from Schedule"***
42. Private Const g_BottlingPlantAll As String = ***"Select * from BottlingPlant"***
43. Private Const g_ExperimentAll As String = ***"Select * from Experiment"***
44. Private Const g_ParettoDominatedChr As String = ***"Select * from ParettoDominatedChr"***
45. Private Const g_ParettoNonDominatedChr As String = ***"Select * from ParettoNonDominatedChr"***
46. Private Const g_TruckAll As String = ***"Select * from Truck"***
47. Private Const g_TruckAllByFirstPosition = ***"SELECT Truck.intTruckId, Route.intRouteId, BottlingPlant.BPId, Truck.intCapacity FROM [Truck INNER JOIN BottlingPlant ON Truck.intCurrentLocation = BottlingPlant.BPId] INNER JOIN Route ON BottlingPlant.BPId = Route.intSourceId"***
48. Private Const g_RndEvaluationAll As String = ***"Select * from RndEvaluation"***
49. Private Const g_RndEvaluationAllBatchRds As String = ***"Select intTruckId, intRouteId, intChromosome, intStartTime, intStartTime1 from RndEvaluation"***

50. Private Const g_RndBpRepairTech As String = ***"SELECT sum[intCapacity] as sumCapacity, intSourceId FROM [RndEvaluation INNER JOIN Route ON RndEvaluation.intRouteId = Route.intRouteId] INNER JOIN Truck ON RndEvaluation.intTruckId = Truck.intTruckId "***
51. Private Const g_RndRefineryRepairTech As String = ***"SELECT sum[intCapacity] as sumCapacity, intDestinationId FROM [RndEvaluation INNER JOIN Route ON RndEvaluation.intRouteId = Route.intRouteId] INNER JOIN Truck ON RndEvaluation.intTruckId = Truck.intTruckId "***
52. Private Const g_RndGetOverSupplyScheduleId As String = ***"SELECT intScheduleId FROM [RndEvaluation INNER JOIN Route ON RndEvaluation.intRouteId = Route.intRouteId] INNER JOIN Truck ON RndEvaluation.intTruckId = Truck.intTruckId "***
53. Private Const g_RndGetStarvationScheduleId As String = ***"SELECT intScheduleId, RndEvaluation.intTruckId, RndEvaluation.intStartTime, RndEvaluation.intRouteId, RndEvaluation.intChromosome FROM [RndEvaluation INNER JOIN Route ON RndEvaluation.intRouteId = Route.intRouteId] INNER JOIN Truck ON RndEvaluation.intTruckId = Truck.intTruckId "***
54. Private Const g_RouteMin As String = ***"SELECT Min[intDistance] AS MinDistance FROM [BottlingPlant INNER JOIN Route ON BottlingPlant.BPId = Route.intDestinationId] INNER JOIN Refinery ON Route.intDestinationId = Refinery.RefineryId "***
55. Private Const g_RndEvalDueDateCost As String = ***"Select RndEvaluation.*, intTime FROM RndEvaluation INNER JOIN route ON RndEvaluation.intRouteId = route.intRouteId"***
56. Private Const g_BestChromosomeDueDateCost As String = ***"SELECT BestChromosome.*, intTime FROM BestChromosome INNER JOIN route ON BestChromosome.intRouteId = route.intRouteId"***
57. Private Const g_RouteComp As String = ***"SELECT Refinery.intSupplyMonth, Route.intRouteId, BottlingPlant.intGasDemandMonth FROM BottlingPlant INNER JOIN [Route INNER JOIN Refinery ON Route.intDestinationId = Refinery.RefineryId] ON BottlingPlant.BPId = Route.intSourceId;"***
58. Private Const g_RouteCompAll As String = ***"SELECT Refinery.txtName as RefName, Route.intRouteId, BottlingPlant.txtName as BPName, intCost FROM BottlingPlant INNER JOIN [Route INNER JOIN Refinery ON Route.intDestinationId = Refinery.RefineryId] ON BottlingPlant.BPId = Route.intSourceId;"***
59. Private Const g_ScheduleComp As String = ***"SELECT Truck.txtTruckName, Refinery.txtName as RefName, BottlingPlant.txtName as BpName, Route.intRouteId, Schedule.intScheduleId, Schedule.intChromosome, truck.IntTruckId, intCost, intDistance, intfatigue, intTime, truck.IntCapacity, RefineryId, BpId FROM [[Route INNER JOIN [Truck INNER JOIN Schedule ON Truck.intTruckId=Schedule.intTruckId] ON Route.intRouteId=Schedule.intRouteId] INNER JOIN Refinery ON Route.intDestinationId=Refinery.RefineryId] INNER JOIN BottlingPlant ON Route.intSourceId=BottlingPlant.BPId"***
60. Private Const g_BestChromosomeCompelte As String = ***"SELECT Truck.txtTruckName, Refinery.txtName as RefName, BottlingPlant.txtName as BpName, Route.intRouteId, BestChromosome.intScheduleId, BestChromosome.intChromosome, truck.IntTruckId, intCost, intDistance, intfatigue, intTime, truck.IntCapacity, RefineryId, BpId, intStartTime FROM [[Route INNER JOIN [Truck INNER JOIN BestChromosome ON Truck.intTruckId=BestChromosome.intTruckId] ON Route.intRouteId=BestChromosome.intRouteId] INNER JOIN Refinery ON Route.intDestinationId=Refinery.RefineryId] INNER JOIN BottlingPlant ON Route.intSourceId=BottlingPlant.BPId"***
61. Private Const g_RndEvalUnionCost As String = ***"Select count[RndEvaluation.intRouteId] as sumRoute, RndEvaluation.intRouteId, intUnionFactor FROM RndEvaluation INNER JOIN route ON RndEvaluation.intRouteId = route.intRouteId"***
62. Private Const g_BestChrUnionCost As String = ***"Select count[BestChromosome.intRouteId] as sumRoute, BestChromosome.intRouteId, intUnionFactor FROM BestChromosome INNER JOIN route ON BestChromosome.intRouteId = route.intRouteId"***
63. Private Const g_RndEvalRoute As String = ***"Select RndEvaluation.* , Route.intTime, Route.intDistance FROM RndEvaluation INNER JOIN route ON RndEvaluation.intRouteId = route.intRouteId"***
64. Private Const g_RndEvalDueDateBatchCost As String = ***"Select RndEvaluation.*, intTime FROM RndEvaluation INNER JOIN route ON RndEvaluation.intRouteId = route.intRouteId"***
65. Private Const g_ScheduleCost As String = ***"SELECT intChromosome, sum[intcost] as sumCost " & " FROM schedule INNER JOIN route ON route.intRouteId=schedule.intRouteId where inttruckId <> 0 group by intChromosome"***

66. Private Const g_FinalTravelTime As String = ***"SELECT sum[intcost] as sumCost, intTruckId, sum[intStartTime] as SumTravelTime FROM RndEvaluation INNER JOIN route ON route.intRouteId=RndEvaluation.intRouteId"***
67. Private Const g_BestChrFinalTravelTime As String = ***"SELECT sum[intcost] as sumCost, intTruckId, sum[intStartTime] as SumTravelTime FROM BestChromosome INNER JOIN route ON route.intRouteId = BestChromosome.intRouteId"***
68. Private Const g_BestChrTotalWorkingHour As String = ***"SELECT Sum[route.intTime] AS sumTravelTime, Sum[route.intSafety] AS sumSaftey, Sum[route.intFatigue] AS sumFatigue, BestChromosome.intChromosome, Truck.txtTruckName, BestChromosome.intTruckId FROM [BestChromosome INNER JOIN route ON BestChromosome.intRouteId = route.intRouteId] INNER JOIN Truck ON BestChromosome.intTruckId = Truck.intTruckId"***
69. Private Const g_BestChrTotalTravelDistance As String = ***"SELECT Sum[route.intDistance * route.intCost] AS sumDistanceCost, Sum[route.intDistance] as sumDistance , BestChromosome.intChromosome, txtTruckName, Truck.intTruckId, Sum[intAccidentNumber] as sumAccidentNumber FROM [BestChromosome INNER JOIN route ON BestChromosome.intRouteId = route.intRouteId] INNER JOIN Truck ON BestChromosome.intTruckId = Truck.intTruckId"***
70. Private Const g_TotalTravelDistance As String = ***"SELECT Sum[route.intDistance * route.intCost] AS sumDistanceCost, Sum[route.intDistance] as sumDistance , Sum[intAccidentNumber] as sumAccidentNumber, intChromosome, intTruckId FROM RndEvaluation INNER JOIN route ON route.intRouteId=RndEvaluation.intRouteId"***
71. Private Const g_TotalWorkingHour As String = ***"SELECT Sum[route.intTime] AS sumTravelTime, Sum[route.intSafety] AS sumSaftey, Sum[route.intFatigue] AS sumFatigue, RndEvaluation.intChromosome, Truck.txtTruckName, RndEvaluation.intTruckId FROM [RndEvaluation INNER JOIN route ON RndEvaluation.intRouteId = route.intRouteId] INNER JOIN Truck ON RndEvaluation.intTruckId = Truck.intTruckId"***
72. Private Const g_TotalOperation As String = ***"SELECT Count[intScheduleId] AS sumOperationTime, RndEvaluation.intChromosome, Truck.txtTruckName, RndEvaluation.intTruckId FROM RndEvaluation INNER JOIN Truck ON RndEvaluation.intTruckId = Truck.intTruckId"***
73. Private Const g_EvaluationCostGroupByChromosome As String = ***"SELECT sum[intcost] as sumCost, intChromosome, max[intStartTime] as SumTravelTime FROM RndEvaluation INNER JOIN route ON route.intRouteId=RndEvaluation.intRouteId where inttruckid <> 0 group by intChromosome"***
74. Private Const g_NumberOfTrucksUsed As String = ***"SELECT RndEvaluation.intTruckId, RndEvaluation.intChromosome From RndEvaluation"***
75. Private Const g_BestChrNumberOfTrucksUsed As String = ***"SELECT BestChromosome.intTruckId, BestChromosome.intChromosome From BestChromosome"***
76. Private Const g_totalOtherCost As String = ***"SELECT sum[intcost] as sumOtherCost FROM RndEvaluation INNER JOIN route ON route.intRouteId = RndEvaluation.intRouteId"***
77. Private Const g_BestChrtotalOtherCost As String = ***"SELECT sum[intcost] as sumOtherCost FROM BestChromosome INNER JOIN route ON route.intRouteId = BestChromosome.intRouteId"***
78. Private Const g_BestChrBetweenDistanceCost As String = ***"SELECT * FROM BestChromosome INNER JOIN route ON route.intRouteId = BestChromosome.intRouteId"***
79. Private Const g_BetweenDistanceCost As String = ***"SELECT * FROM RndEvaluation INNER JOIN route ON route.intRouteId = RndEvaluation.intRouteId"***
80. Private Const g_BPInventoryCost As String = ***"SELECT count[Route.intSourceId] as deliveredLpg, BottlingPlant.txtName, BottlingPlant.intCapacity, BottlingPlant.intStarvMonthFactor, BottlingPlant.intStarvationMonthly, BottlingPlant.intOverSupplyFactor, BottlingPlant.intOverSupply FROM [RndEvaluation INNER JOIN Route ON RndEvaluation.intRouteId = Route.intRouteId] INNER JOIN BottlingPlant ON Route.intSourceId = BottlingPlant.BPId"***
81. Private Const g_BestChrBPInventoryCost As String = ***"SELECT count[Route.intSourceId] as deliveredLpg, BottlingPlant.txtName, BottlingPlant.intCapacity, BottlingPlant.intStarvMonthFactor, BottlingPlant.intOverSupplyFactor, BottlingPlant.intStarvationMonthly, BottlingPlant.intOverSupply FROM [BestChromosome INNER JOIN Route ON BestChromosome.intRouteId = Route.intRouteId] INNER JOIN BottlingPlant ON Route.intSourceId = BottlingPlant.BPId"***
82. Private Const g_RefineryInventoryCost As String = ***"SELECT count[Route.intDestinationId] as utilised, Refinery.txtName, Refinery.intCapacity, Refinery.intAllowMonthFactor, Refinery.intAllowDayFactor FROM [RndEvaluation INNER JOIN Route ON RndEvaluation.intRouteId = Route.intRouteId] INNER JOIN Refinery ON Route.intDestinationId = Refinery.RefineryId"***

83. Private Const g_BestChrRefineryInventoryCost As String = ***"SELECT count[Route.intDestinationId] as utilised, Refinery.txtName, Refinery.intCapacity, Refinery.intAllowMonthFactor, Refinery.intAllowDayFactor FROM [BestChromosome INNER JOIN Route ON BestChromosome.intRouteId = Route.intRouteId] INNER JOIN Refinery ON Route.intDestinationId = Refinery.RefineryId"***

C.5 Scaling Methods

The following approaches are alternative scaling methods developed for this GA application.

C.5.1 Linear Transformation Approach:

This is a useful scaling procedure introduced in [C-1]. In this method the raw fitness is defined as f and the scaled fitness as $f^{'}$. Linear scaling requires a linear relationship between $f^{'}$ and f as follows:

$$f^{'} = af + b$$

The coefficient ***a*** and ***b*** can be determined in different way. However the aim is to keep the average scaled fitness to be equal to the average raw fitness. This is to ensure that each average population member contributes one expected offspring to the next generation. To control the number of offspring given to the population member with maximum raw fitness the following equation can be used.

$$f^{'}_{\max} = C_{mult} \times f_{avg}$$

C_{mult} is the number of expected copies desired for the best population member. For a typical small population (n = 50 to 100) this value is mostly set to 1.2 to 2. This approach is conducted in [C-1] using three functions namely, ***prescale***, ***scale*** and ***scalepop*** as follows:

- **Prescale Function:**
- This procedure takes the average, maximum, and a minimum raw fitness value, called ***umax, uavg,*** and ***umin***, and calculates linear scaling coefficients ***a*** and ***b*** as described earlier. If it is possible to scale to the desired multiple, $C_{\max}$ (known as ***fmultiple*** in the code), then that is the computation performed. Otherwise, scaling is performed by pivoting about the average value and stretching the fitness until the minimum value maps to zero.

```
Function Presacle (umax, uavg, umin: real; var a, b: real);
Const fmultiple = 1.5     {fitness multiple is set to 1.5}
Var delta: real;      {devisor}
Begin
 If umin> (fmultiple*uavg-umax)/(fmultiple-1)  { none negative test}
  then begin { Normal Scaling}
  Delta:= umax-uavg;
  a := (fmultiple-1)*uavg/delta;
  b := uavg* (umax-fmultiple*uavg)/delta;
  end else begin     { Scale as much as possible}
  Delta:= uavg-umin;
  a := uavg/delta;
  b :=-umin*uavg/delta;
 end;
end;
```

Figure C-28 Pre-Scale Function

- **Scalepop Function:**
- This function is called after ***prescale*** procedure to scale all the individual raw fitness values using the simple function ***scale*** as described in the following.

```
Function scale (u, a, b: real): real;
 Begin
 scale: = a* u + b
end
```

Figure C-29 Scale Function

```
Function Scalepop (popsize: integer; var max, avg, min, sumfitness: real; var pop: population);
{Scale entire population}
var j: integer;
a, b:= real; {slope & intercept for linear equation}
begin
prescale (max, avg, min, a, b); { get slop and intercept for function}
sumfitness:=0;
 for j:=1 to popsize do with pop[j] do begin
  fitness: = scale (objective, a, b);
  sumfitness;= sumfitness + fitness;
 end;
end;
```

Figure C-30 Scaleop function

Using the fitness values obtained in section 4.5.2 and using the above procedures the linear scaling of fitness values is obtained as follows:

Chromosomes	Raw fitness Values	Scaled Fitness value Linear Transformation
1	6758	25967.3
2	52560	48868.3
3	26786	35981.3
4	106141	75658.8
5	101384	73280.3
6	107390	76283.3
7	135709	90442.8
8	112194	78685.3
9	0	22588.3
10	105216	75196.3

Figure C-31 Scaling using Linear Transformation

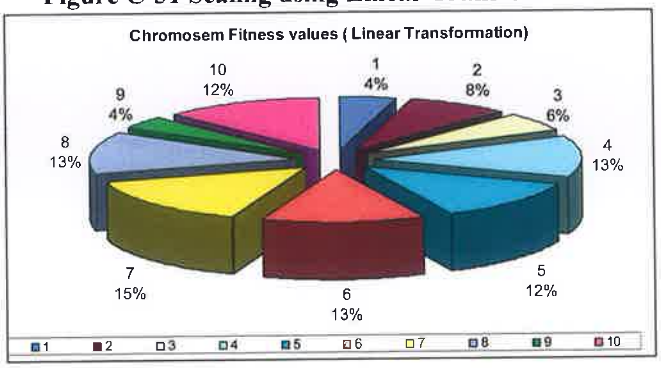


Figure C-32, linear scaling mechanism

C.5.2 Ranked Based Fitness Assignments

In this method the population is sorted according to their cost values. The fitness assigned to each individual depends only on it position in the individual rank and not on the actual cost values. This method attempts to prevent possible premature convergence, when the selective pressure has cause the search to narrow down too quickly. This method introduces a uniform scaling across the population and provides a simple and effective way of controlling selective pressure. Considering ***Nind*** to represent the number of individual chromosomes in the population, Pos the position of an individual in this population (the least fit individual has ***Pos*** =1, the fitness chromosome ***Pos***= ***Nind***) and ***SP*** the selective pressure. The fitness value for a chromosome is calculated as follows [C-1]:

$$\textbf{Fitness}\ (Pos) = 2 - SP + 2.\ (SP - 1).\ (Pos - 1) / (Nind - 1)$$

Typical values for SP are in the range of [1.0, 2.0]. Figure C- **33**, illustrates how this approach was coded.

```
Private Function ScalingRankedLinearMethod() As Double
  isAdded = False
  Dim valElement As Variant
  ' calc sum of all fitness values
  For Each valElement In dicFitness.Keys
   ' to rank the primary costs
    For j = 0 To colRevRanked.count - 1
      If (colRevRanked(j + 1) < dicFitness.Item(valElement)) Then
        Call colRevRanked.Add(dicFitness.Item(valElement), , j + 1)
        isAdded = True
        Exit For
      End If
    Next
    If (isAdded = False) Then
      colRevRanked.Add dicFitness.Item(valElement)
    End If
    isAdded = False
  Next
  ' to obtain the fitness of each individual in a generation
  For i = 1 To colRevRanked.count
    afterScaled = LinearRankedScalingFitness(i, colRevRanked.count)
    iRealChromosomePosition = GetIndexOfDictionary(dicFitness.Items,
colRevRanked(i))
    dicScaledFitness.Add iRealChromosomePosition, afterScaled
    sumTotalFitnesses = sumTotalFitnesses + afterScaled
  Next
  ' return the total scaled chromosome
  ScalingRankedLinearMethod = sumTotalFitnesses
End Function
```

Figure C- 33, Ranked based fitness assignment method

C.6 Selection Method

The tournament method is an alternative selection method developed in this application. What fallows demonstrates how this method was implemented in this application.

C.6.1 Tournament:

In this selection, pairs of individuals are picked at random from the population. The one with higher fitness is copied into a new pool for mating operations. This process is repeated as often as individuals must be chosen. Also, larger tour sizes are employed by picking ***n*** individuals instead of two. This increases the selection pressure. The fitter member in the pool has a higher chance to get selected for reproduction. Figure C-34 illustrates the sequence involved in this selection method and Figure C-35 illustrates how this method was implemented. There are several advantages in using tournament selection [C-1]:

- This approach is shown to work well and superior to roulette wheel.
- The tournament size can vary.
- This method could be extended to multiple criteria problems.

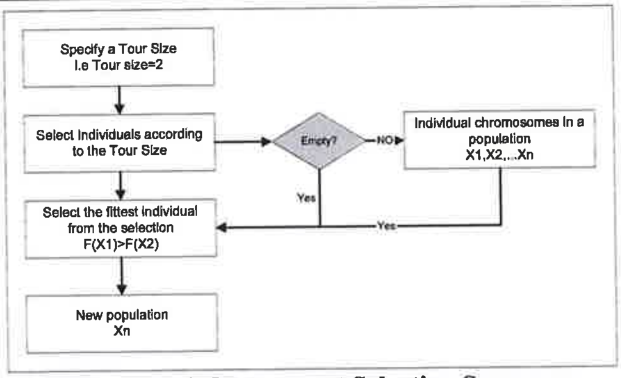


Figure C-34 Tournament Selection Sequence

```
Private Function SelectionTournamentMethod() As Integer

  Dim tournamentPt As Integer
  Dim bestChrNumber As Integer
  Dim mostProfitable  As Double
  Dim scaledChromosome As Double
  Dim i As Integer

  tournamentPt = GetRndNumber(1, GASpec.tournamentNo)
  mostProfitable = dicScaledFitness.Items(0)
  bestChrNumber = 1
  Dim valElement As Variant

  ' calc sum of all fitness values
  For i = 0 To (dicScaledFitness.count) - 1
    scaledChromosome = dicScaledFitness.Item(dicScaledFitness.Keys(i))
    If (scaledChromosome > mostProfitable) Then
      mostProfitable = scaledChromosome
      bestChrNumber = dicScaledFitness.Keys(i)
    End If
  Next
  SelectionTournamentMethod = bestChrNumber
End Function
```

Figure C-35, Tournament selection method

C.7 Crossover operators

In chapter 4 in section 5.8.7 a single point fixed length crossover operator was discussed. Here two other alternative crossover methods developed are illustrated.

C.7.1 Single Point, Variable Length Crossover:

This crossover operator results in an offspring that may have variable number of trucks. In this case a crossover point is selected by generating a random number between the minimum and maximum of the shortest chromosome as in the last case. Then the first portions stay fixed and the second portions are exchanged.

Parent Chromosomes												
	1	2	3	4	5	6	7	8	9	10	11	12
			Crossover									
	Min						Max					
CH1	1	2	3	4	5	6						
CH2	A	B	C	D	E	F	G	H	I	J	K	L

Figure C-36 crossover operation

	Offspring Chromosomes											
	1	2	3	4	5	6	7	8	9	10	11	12
Ofs1	1	2	3	D	E	F	G	H	I	J	K	L
Ofs2	A	B	C	4	5	6						

Figure C-37 variable length offspring generated.

Figure C-38 and Figure C-39 illustrate variable length offspring and their corresponding Fitness's generated using this crossover operator. Figure C-40, represents how this method was implemented.

Cost	S1	CL	SL	EL	SL	EL	SL	EL	SL	EL	SL	EL
78954			27	27	15	15	19	19	28	28	15	15
7695	T1	Babol	Esfahan	Karaj	Abas	Qom	Arak	Karaj	Esfahan	Kashan	Abas	Qom
			12	12	14	14	11	11	31	31		
13675	T2	Esfahan	Abas	Kashan	Abas	Mashhad	Abas	Karaj	Esfahan	Qom		
			35	35	12	12	18	18	9	9	11	11
15765	T3	Karaj	Tehran	Karaj	Abas	Kashan	Arak	EsfahanC	Abas	Babol	Abas	Karaj
			15	15	7	7	19	19	14	14		
13490	D	Esfahan	Abas	Qom	Abadan	Qom	Arak	Karaj	Abas	Mashhad		
			29	29								
10054	E	Karaj	Esfahan	Kerman								
			2	2	14	14	14	14				
18275	F	Kashan	Abadan	EsfahanC	Abas	Mashhad	Abas	Mashhad				
			3	3	8	8	35	35	10	10		
26821	G	Kerman	Abadan	Karaj	Abadan	yazd	Tehran	Karaj	Abas	EsfahanC		
			26	26	11	11	37	37				
8400	H	Mashhad	Esfahan	EsfahanC	Abas	Karaj	Tehran	Kerman				
			25	25	7	7	38	38	22	22		
12380	I	Qom	Esfahan	Babol	Abadan	Qom	Tehran	Mashhad	Arak	Mashhad		
			23	23	12	12	33	33				
17960	J	Yazd	Arak	Qom	Abas	Kashan	Tehran	Babol				
			33	33	26	26	17	17	10	10		
6406	K	Abadan	Tehran	Babol	Esfahan	EsfahanC	Arak	Babol	Abas	EsfahanC		
			13	13	2	2	21	21	19	19	35	35
10125	L	Abas	Abas	Kerman	Abadan	EsfahanC	Arak	Kerman	Arak	Karaj	Tehran	Karaj

Figure C-38 Variable length offspring1

Cost	S2	CL	SL	EL	SL	EL	SL	EL	SL	EL	SL	EL
67348			38	38	13	13						
19763	A	Esfahan	Tehran	Mashhad	Abas	Kerman						
			3	3	39	39	17	17				
11496	B	Tehran	Abadan	Karaj	Tehran	Qom	Arak	Babol				
			4	4	21	21	27	27				
6738	C	Babol	Abadan	Kashan	Arak	Kerman	Esfahan	Karaj				
			23	23	18	18	22	22				
15150	T4	Kashan	Arak	Qom	Arak	EsfahanC	Arak	Mashhad				
			14	14	15	15	15	15	40	40		
14201	T5	Kerman	Abas	Mashhad	Abas	Qom	Abas	Qom	Tehran	yazd		
			12	12	22	22	24	24	11	11		
13131	T6	Mashhad	Abas	Kashan	Arak	Mashhad	Arak	yazd	Abas	Karaj		

Figure C-39 Variable length offspring 2

```
Private Sub DoCrossOverVariableLength(ByRef rdsChr1st As ADODB.Recordset,
ByRef rdsChr2nd As ADODB.Recordset, _
        iSplitPt As Integer, ByRef col1st As Collection, ByRef col2nd As Collection)

   Dim tmpSchdule As clsScheduleData
   Dim coltmpChr1 As Collection
   Dim coltmpChr2 As Collection
   Dim firstChr As Integer
   Dim secondChr As Integer

   rdsChr1st.MoveFirst
   rdsChr2nd.MoveFirst

   firstChr = rdsChr1st.Fields("intChromosome")
   secondChr = rdsChr2nd.Fields("intChromosome")
   ' obtain the data before split point and save them to the suitable collection
   Set col1st = FillCollectionBestChrs(rdsChr1st, rdsTrucksNoChr1, True, iSplitPt)
   Set col2nd = FillCollectionBestChrs(rdsChr2nd, rdsTrucksNoChr2, True, iSplitPt)
   rdsTrucksNoChr1.MoveFirst
   rdsTrucksNoChr2.MoveFirst
   If (rdsTrucksNoChr1.EOF) Then Exit Sub
   If (rdsTrucksNoChr2.EOF) Then Exit Sub
   ' obtain the data after split point and save them to the suitable collection
   Set coltmpChr1 = FillCollectionBestChrs(rdsChr1st, rdsTrucksNoChr1, False,
iSplitPt)
   Set coltmpChr2 = FillCollectionBestChrs(rdsChr2nd, rdsTrucksNoChr2, False, iSplitF
' chromosomes change their data placed after split point
   Call DoTraverse(coltmpChr1, coltmpChr2, colAfterChange1, colAfterChange2)

   ' add new exchanged data to the built collection(above) built before split point
   ' in order to get the complete chromosomes
   Call AddToCollection(colAfterChange1, col1st)
   Call AddToCollection(colAfterChange2, col2nd)
End Sub
```

Figure C-40, Variable Length Crossover Method

C.7.2 Partially Mapped Uniform Crossover:

In this operator, at the top level, the smallest chromosome is selected. Two genes are randomly selected from the minimum and maximum number of trucks exists in the chromosome. Here genes refer to individual truck schedule. The corresponding genes are obtained from each individual parent chromosome. Establishing the minimum and maximum length of the genes, a random number is generated (x). The first offspring is obtained by copying the first gene. Randomly (x) fractions of this gene are selected. These fractions are replaced with the corresponding genes from the second gene. The modified gene is replaced back into the first chromosome resulting in a new offspring. This procedure is also followed on the second parent chromosome to generate the second offspring. The following graphically demonstrated this crossover mechanism. Figure C-45 demonstrate how this method was implemented in this application.

- Random selection of genes from top level parent chromosomes

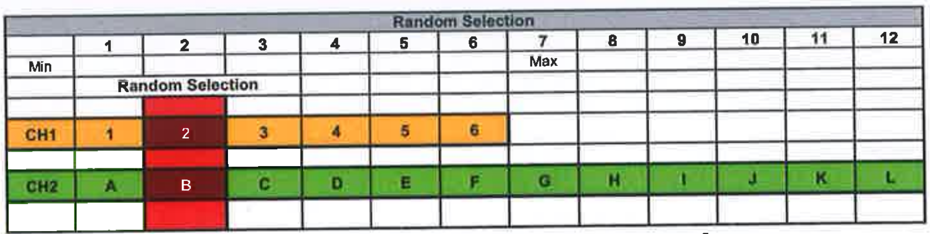


Figure C-41 Random Selection of genes from parent chromosomes

- Selection of two individual genes from the original parent chromosomes. Randomly a number of fractions are chosen from a smaller gene. In this case the random number is generated between one and three (e.g., rand [1,2] = 2) as there are only three routes in gene B.

Individual Genes												
	1	2	3	4	5	6	7	8	9	10	11	12
	Cost			12	12	14	14	11	11	31	31	
Gene1	13675	T2	Esfahan	Abas	Kashan	Abas	Mashhad	Abas	Karaj	Esfahan	Qom	
				3	3	39	39	17	17			
Gene2	11496	B	Tehran	Abadan	Karaj	Tehran	Qom	Arak	Babol			
				Min					Max			
				1		2		3				

Figure C- 42 Selection of Individual genes from parent chromosomes

- Modification of the first gene by exchanging routes with the chosen fractions from the second gene.

Modifcation of the 1st Gene												
	1	2	3	4	5	6	7	8	9	10	11	12
				3	3	39	39	11	11	31	31	
CH1	12980	T2	Esfahan	Abadan	Karaj	Tehran	Qom	Abas	Karaj	Esfahan	Qom	

Figure C- 43 Modification of the First Gene

- Updating the first parent chromosome with the modified gene to obtain the new offspring.

Cost	S1	CL	SL	EL	SL	EL	SL	EL	SL	EL	SL	EL
78922			27	27	15	15	19	19	28	28	15	15
7695	T1	Babol	Esfahan	Karaj	Abas	Qom	Arak	Karaj	Esfahan	Kashan	Abas	Qom
			3	3	39	39	11	11	31	31		
12980	T2	Esfahan	Abadan	Karaj	Tehran	Qom	Abas	Karaj	Esfahan	Qom		
			35	35	12	12	18	18	9	9	11	11
15765	T3	Karaj	Tehran	Karaj	Abas	Kashan	Arak	EsfahanC	Abas	Babol	Abas	Karaj
			15	15	7	7	19	19	14	14		
15150	T4	Kashan	Arak	Qom	Arak	EsfahanC	Arak	Mashhad				
			14	14	15	15	15	15	40	40		
14201	T5	Kerman	Abas	Mashhad	Abas	Qom	Abas	Qom	Tehran	yazd		
			12	12	22	22	24	24	11	11	10	
13131	T6	Mashhad	Abas	Kashan	Arak	Mashhad	Arak	yazd	Abas	Karaj	Abas	

Figure C-44 Offspring 1: updating the first chromosome

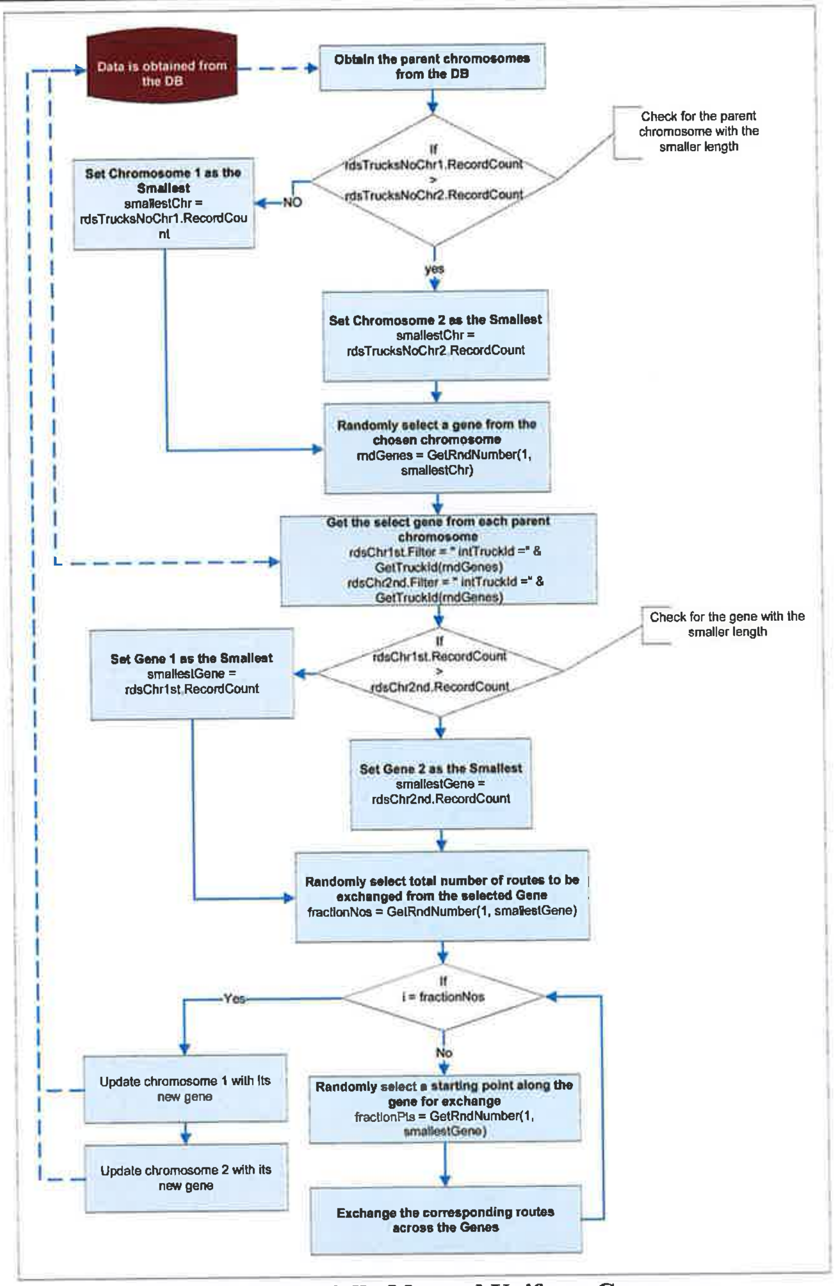


Figure C-45 Partially Mapped Uniform Crossover

C.8 Mutation Operator

Chapter 4 section 5.8.8 described the mutation operators developed for this application. The following is the second alternative method developed for this application

C.8.1 Inversion Mutation

Using this mechanism, two points are chosen along the length of the chromosome. Then the chromosome is cut at these points and the end points of the cut section switch places. For example considering the following nine position chromosome, where two inversion sites are chosen at random (e.g. site 3 and 7). Figure C-47 shows how this method is implemented and it indicate its impact on the chromosome cost. Figure C-48, illustrates how this operator was coded.

Cost		1	2	3	4	5	6	7	8	9
			12	12	22	22	24	24	11	11
13131	T6	Mashhad	Abas	Kashan	Arak	Mashhad	Arak	yazd	Abas	Karaj

Cut Point1 Cut Point2

Figure C-46 A gene before Inversion

Cost		1	2	3	7	6	5	4	8	9
			12	12	24	24	22	22	11	11
12900	T6	Mashhad	Abas	Kashan	Arak	yazd	Arak	Mashhad	Abas	Karaj

Figure C-47 a gene after Inversion mutation

```
Private Sub MutationInversion(ByRef rds As ADODB.Recordset)
    Dim I As Integer
    Dim rndRoute As Integer
    Dim colOneTruckSchdl As New Scripting.Dictionary
    Dim objTemp As Variant

    With rds
      .MoveFirst
      ' 1 for first location of truck

      For I = 0 To .RecordCount - 1
        colOneTruckSchdl.Add I, .Fields("intRouteId").Value
        .MoveNext
      Next

      For i = 0 To colOneTruckSchdl.count - 1
         .MoveFirst
         .Move i
         If (.EOF) Then Exit For
         rndRoute = GetRndNumber(1, colOneTruckSchdl.count - 1)
         Do While Not (colOneTruckSchdl.Exists(rndRoute))
            rndRoute = GetRndNumber(1, colOneTruckSchdl.count)
         Loop
         .Fields("intRouteId") = colOneTruckSchdl.Items(rndRoute)
         'colOneTruckSchdl.Remove (rndRoute)
      Next

      .ActiveConnection = cEval.GetActiveConnectionBatchRds
      .UpdateBatch adAffectAllChapters
    End With
End Sub
```

Figure C-48, Inversion Mutation operator

C.9 Replacement Strategies

Chapter 4, section 4.5.9 described how elitism was implemented in this application. The aim of this section is to demonstrate the steps and functions used in ***MakeEliteSchedule (ByRef colCost As Collection)*** procedure to perfume elitism concept in this application.

This function is described in 4 sections. The first function is used to update Elite data collections from competitive chromosomes in each generation. There are two main Elite collections namely known as ***m_ColEliteChromosomes*** and ***colCopyElite***. The latter is a copy of the first collection data. The first collection data is updated in initial population first. The second collection data is updated with the fittest chromosomes form the current generations. Function 1 as illustrated in Figure C-49 performs this operation.

As there might be scenarios, where ***m_ColEliteChromosomes*** is not updated with the latest fit chromosomes. A new dada collection known as ***CandidateCol*** is generated. In this collection the best chromosomes are gathered from both ***m_ColEliteChromosomes*** and ***colCopyElite*** elite collections. This done as shown in Figure C-50

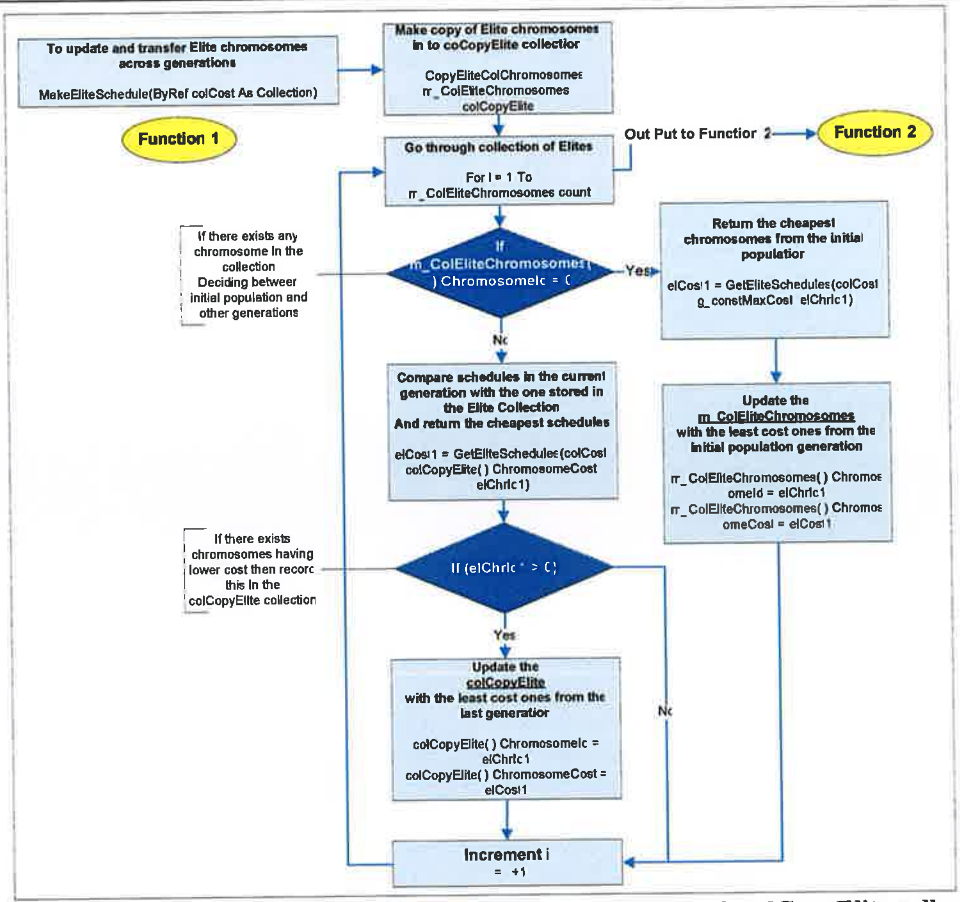


Figure C-49 Function1, used to update m_ColEliteChromosomes & colCopyElite collection

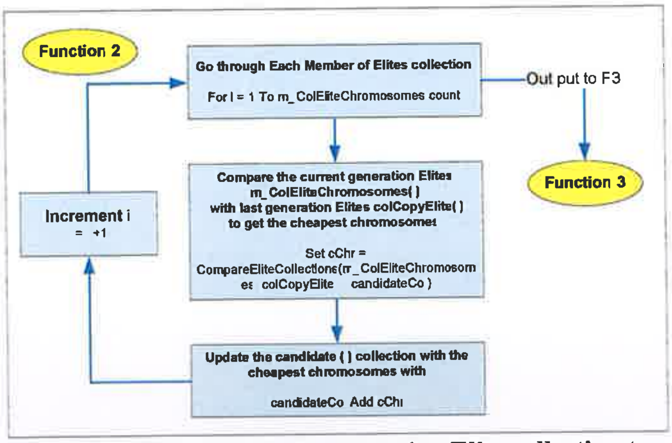


Figure C-50, Function 2, comparing Elite collection to update Candidate data collection

The Candidate collection contains the fittest chromosomes and the most Elite members. Comparing these chromosomes with the members of ***m_ColEliteChromosomes***, the least competitive members of this elite collection are identified. Function 3 as indicated in Figure C-51 is used to remove the least competitive member of elite collection and also it updates the Elitism database with the latest changes.

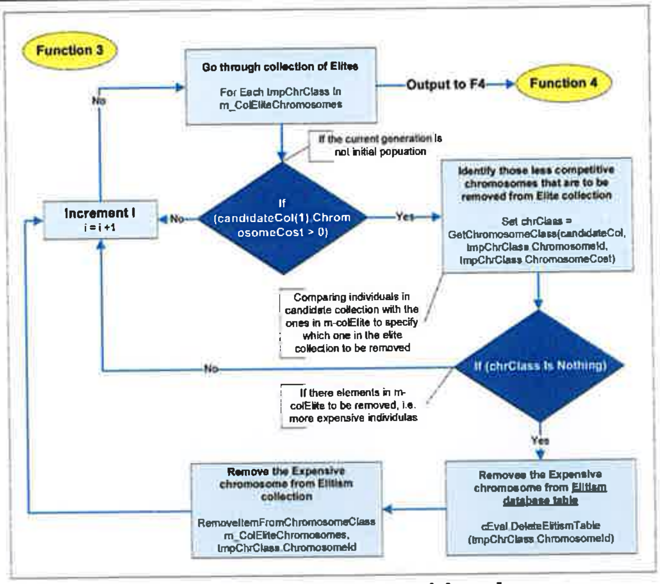


Figure C-51, Removing less competitive chromosomes from elite collection and database

One the lease competitive members are removed a check is done to make sure that the chromosome ids in the Candidate collection and the individuals in ***m_ColEliteChromosomes*** are not the same. This could happen due to the way that databases operate. If the current member of the ***m_ColEliteChromosomes*** has the same id as the current member of the Candidate collection, then a new Id is generated for the chromosome in the Candidate collection. This is done to make sure that each chromosome has its own specific ids and to prevent any further complications. Once the chromosomes' ids are different then the ***m_ColEliteChromosomes*** is updated with the fittest chromosomes. Then the Elite database is updated with the fittest chromosomes. The individual member of Elite database is then transferred to the next generation. Function 4 as illustrated in Figure C-52is used to perform these operations.

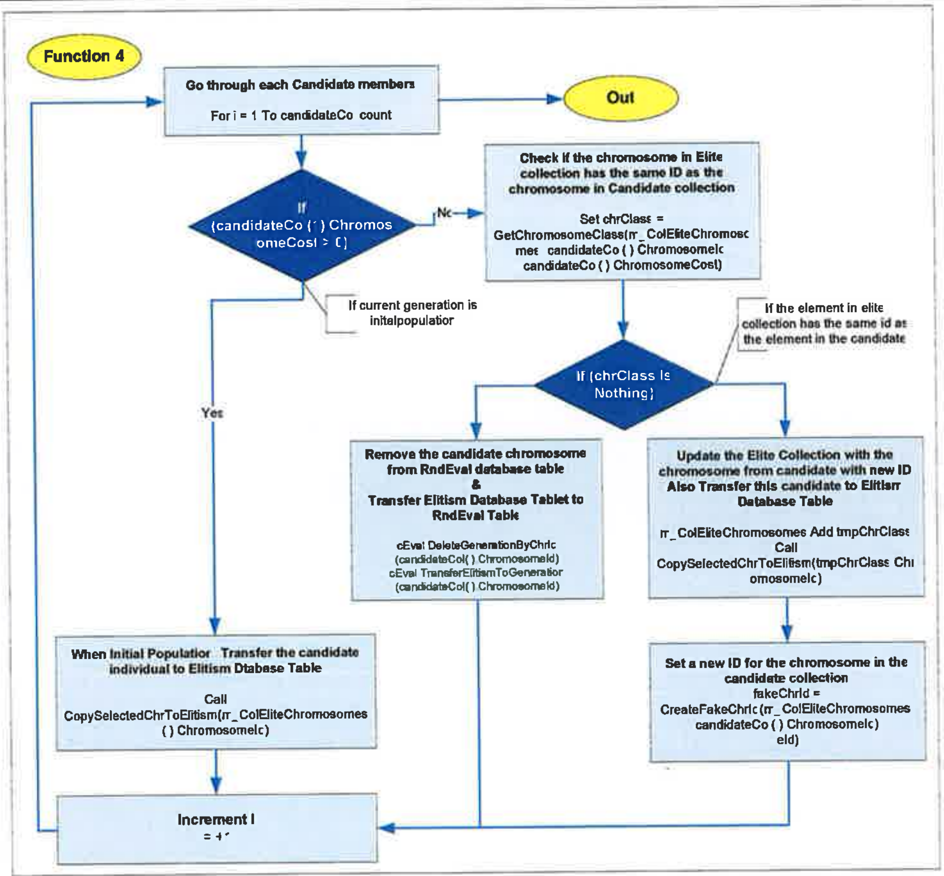


Figure C-52, Function 4, checking the elite chromosome's ids, transferring elite to next generation

C.10 Steps developed in Pareto Optimal Genetic Search Engine

The Following steps are developed to model Pareto Optima Search for this application.

C.10.1 Initialisation:

This step is no different as the initial population described in section **4.5.3**, where a number of individual chromosomes are randomly generated form the main decision variables based on the specified decision spaces. The function called ***DoParettoSelectionMethod*** is used to initiate the Pareto search. Figure C-53, shows the steps in this algorithm. This function starts with initialising arrays to collect dominated and non-dominated individuals ***(m_colNonDominated, m_colDominated)***. It performs dominance check on all members of a population. This is done using function ***TotalDominanceCheck (popNo, genNo)***. This function is a complex nested function, which takes two input arguments such as the population size and the number of generation specified by the user. The component of this function is described latter. The outputs of this function are those non-dominated schedules, which are stored in a collection array.

The dominated individuals are deleted from the population. Then the total numbers of non-dominated schedules are checked to see if it matches the specified population size. If this is not true, new offspring are created from the existing non-dominated chromosomes. In a case, where there is only one non-dominated chromosome; a new chromosome is generated randomly. The selected individuals undergo genetic reproduction process. As a result the recently generated individuals are added to the population. Once again a check is done to remove the non-dominated chromosomes. These steps are carried by a function called ***ParettoGenerateNewSchedule***. This function is further described in C.10.3.

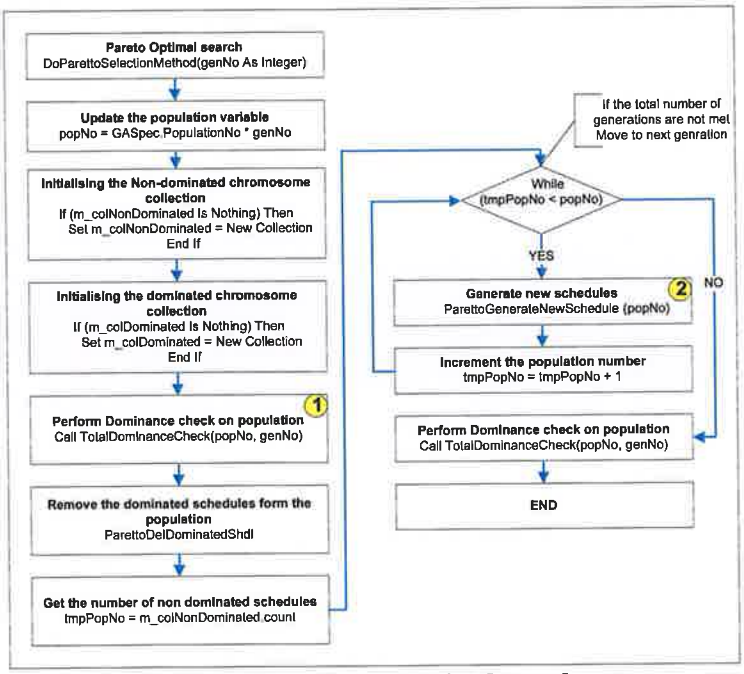


Figure C-53 Pareto Optimal search

C.10.2 Dominance Check

As specified above, Function ***TotalDominanceCheck (popNo, genNo)*** perform the dominance check operation. In this step the current population is checked to see which individuals are non-dominated. Figure C-54, shows the steps taken to check the dominance of individuals in a population. This function first start with going through each individual in a population and record the related cost factors in a collection array ***(m_colSchedulesCos)***. Then the Boolean parameter Dominance for each individual is set to a default ***False*** value. After doing so, the major function called ***DominanceCheckForOneSchedule*** is called. This function is used to compare each individual's cost parameters with other members cost parameters for dominance. This is a complex function as show in Figure C-56.

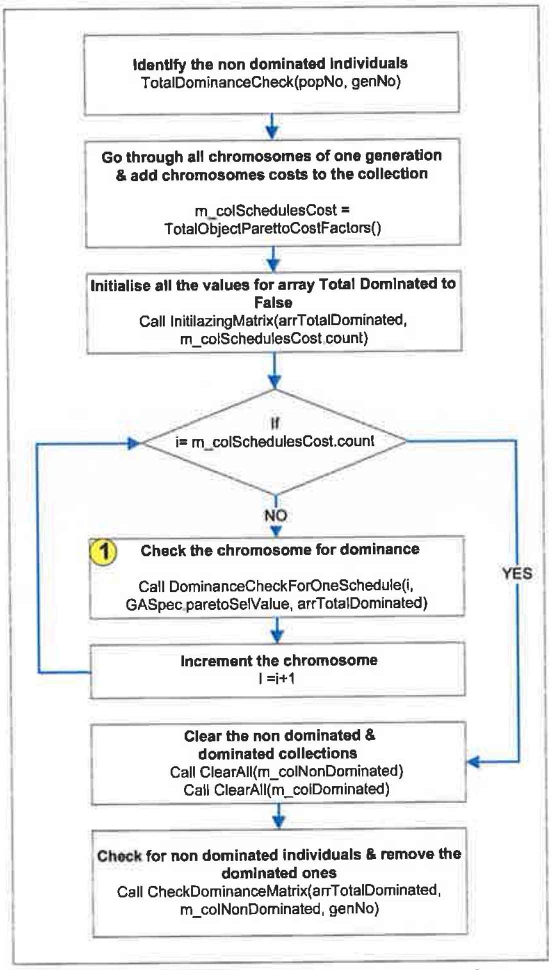


Figure C-54 Dominance Check function

The ***DominanceCheckForOneSchedule*** function starts by reading in the first chromosome from the ***(m_colSchedulesCost)*** collection array. This scheduled is recorded in a collection array known as ***hostScheduleCost***. Further, the algorithmstarts from the beginning of the ***(m_colSchedulesCost)*** array and if the ***hostScheduleCost*** is not the same as the chosen schedule then this schedule is recorded in an array known as ***comparedScheduleCost***. These two arrays are compared for dominance. Here a function is referenced that goes through each individual cost function for each chromosome and compare the relevant costs factors for dominance. This function is called ***ParetoDominanceCheck***.

As shown in Figure C-55, this method is a Boolean function that determines if the host schedule is dominated or not. The inputs to this function are such as the cost objective to be considered, the same cost objective from each considering schedule known as hostScheduleCost and comparedScheduleCost. For instance considering minimisation of distance cost is coded as follows:

ParetoDominanceCheck (hostScheduleCost.DistanceCost, comparedScheduleCost.DistanceCost, paretoObjective.DistanceCost)

In this way, the distance cost from the ***hostSchedule*** is compare with the distance cost from the ***comparedSchedule***. As the objective is to minimise, if the host schedule cost is less than or equal to the compared schedule then output of this function would be ***False*** to indicate the host schedule is not dominated and therefore is non-dominated.

This output is added to a collection array known as ***isDominated*** . This collection is used to hold the output of dominance check for each cost parameter considered. After updating the ***isDominated*** collection with the true and false values; if any of the cost parameters of the Host schedule dominates the other schedule's cost parameter then the host schedule is non-dominated and this schedule is recorded in the non-dominated collection array. Otherwise, it is recorded in the dominated collection array for deletion from the population.

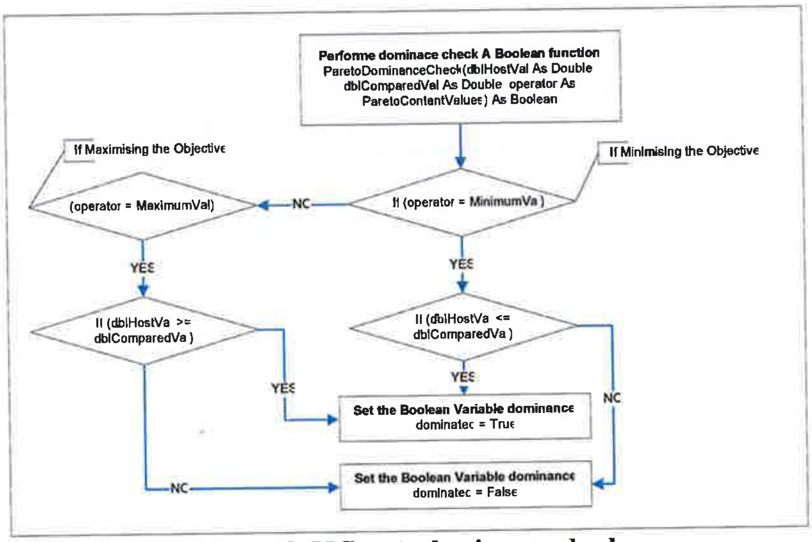


Figure C-55 Pareto dominance check

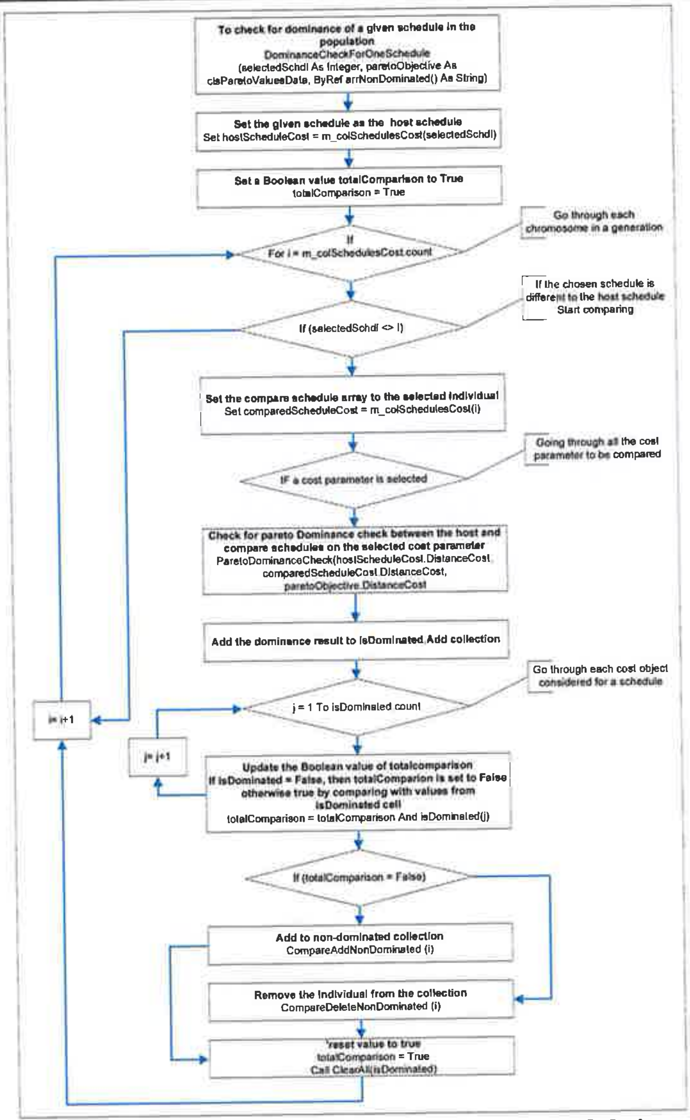


Figure C-56 Checking dominance of each schedule in a population

C.10.3 Adjustment

As mentioned above after the dominance check the dominated individuals are removed from the population. These individuals are replaced with new offspring using the genetic reproduction processes. Figure C-57 shows the steps involved in function ***ParettoGenerateNewSchedule***, to perform this operation.

Using this function, the population is checked to see if all the members are non-dominated individuals, if so, the size of the population is increased and a new offspring is generated by choosing two parents from the non-dominated population. In contrary, if not all the members are non-dominated, then the same procedure is followed until all the members are non-dominated. These steps are followed until the number of generation specified is reached.

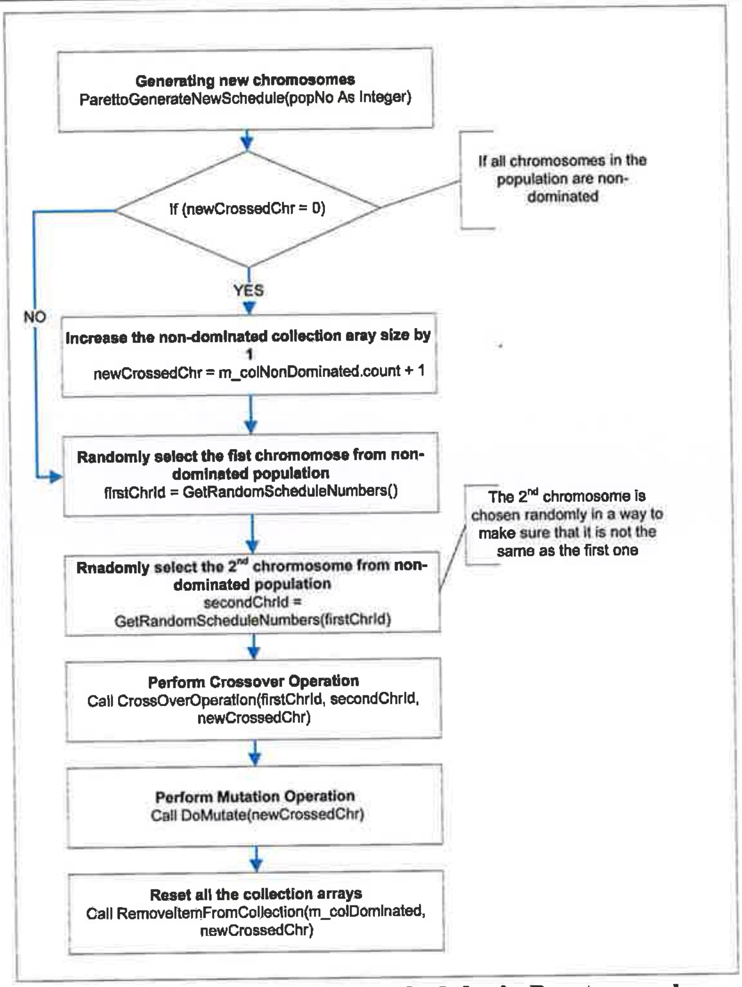


Figure C-57 Generate New schedules in Pareto search

Appendix D: Survey on Simulation Modelling

D.1 Advantages and disadvantages of simulation

Some of the competitive advantages of simulation modelling can be summarised as follows [D-1]:

- The basic component of simulation is easy to comprehend and hence often easier to justify to management or customers than some of the analytical models.
- A simulation model may be more credible because its behaviour has been compared to that of the real system and it captures more of true characteristics of the system.
- Simulation allows experimentation with a model of a system and it prevents potential disruption in a real system and therefore preventing critical risk takings.
- Simulation allows identification of problems, bottlenecks and design shortfalls before building or modifying a system.
- It allows evaluations and comparison of many alternatives designs and rules of operation before committing resources and investment to a project.
- Simulation allows study of the dynamics of a system how it changes over time and how subsystems and components interact.
- A simulation model provides about the only method to study new, non-existent complex dynamic systems for which analytic or static (i.e. spreadsheet) models provide at best a low fidelity model with correspondingly low accuracy.

However in implementing simulation modelling there are a number of drawbacks, which are mentioned as follows [D-1]:

- Simulation modelling requires special training and skills. The quality of the analysis depends on the quality of the model.
- Simulations are time consuming activity and expensive process. Usually data is not available or costly to obtain, and the time available before decisions must have made is not sufficient for a reliable study.
- Simulation results may be difficult to interpret. In some situations, the simulations and other visual displays, combined with the time pressure present on all projects, may mislead decision makers into premature conclusions based on insufficient evidence.
- In addition, inexperienced simulation analyst may spend too much detail to a model and spend substantial time in model development and violating the project timelines consequently. This often leads management to conclude that simulation, while a promising and interesting technology, is too costly and time consuming for most projects.
- Using simulation before a specific problem is articulated may lead to a large number of unfocused simulation runs that used inappropriately designed models, and produce little or no information of value [D-2].

D.2 Simulation Tools:

Jerry Banks et al. [D-3] classified simulation languages at three different levels. These levels are such as:

D.2.1 System

A language can be classified on the basis of the type of system modelled. The two types of systems that are generally recognised are discrete and continuous as explained earlier. In discrete systems, the state variables change only at discrete points in time. These times were referred to previously as event times. Systems involving inventory operations, many manufacturing operations, and material handling systems are usually regarded as discrete systems.

D.2.2 Application

Discrete event simulation software can be classified as special purpose or general purpose. Special purpose simulation languages are designed to model specific environments. These languages offer speedier model development and they are called simulator. This is in contrast to general-purpose simulation software products, which are commonly called a simulation language. The following table indicates the advantages and disadvantages of each approach [D-4].

Table D-9 Simulation Packages vs. General-Purpose Languages

	Advantages	Disadvantage
Simulation Packages	Power software tailored to the purpose of simulation,	Purchase and Training costs.
	It allows user to concentrate on logic of system to be modelled not on computer programming issues.	May not be flexible enough to meet the special features of user's system.
General Purpose Languages	Avoid purchasing costs	Time to develop the program
	Programmers knows the details of the written codes	The entire program as to be debugged
		Learning curve effects

D.2.3 Structural

The structural level is concerned with the modelling perspective taken by the discrete event language. The three most prevalent orientations are as follows Nance [D-5]:

- **Event Scheduling:** In this perspective, a system being modelled is viewed as consisting of a number of possible events at which states changes take place. The modeller defines the events and develops the program logic associated with each event. This approach was used in packages written in FORTRAN. SIMON, SLAM and SIMAN are simulation of this type.
- **Process interaction:** [D-6] this perspective allows a modeller to represent a system as a set of processes. For instance, a part following through several workstations is an example of a process. Another name used to represent process interaction simulation languages is network simulation languages. SIMULA and SIMSCRIPT are typical of this approach.

- **Activity Scanning:** in this perspective, the modeller defines the conditions necessary to start and end each activity and delay in the system. An activity is duration of specified length and a delay is duration of unspecified increments, and if the conditions are appropriate an activity will be started or terminated.

Although a simulation model can be built using general purpose programming languages which are familiar to the analyst, available over wide variety of platforms and less expensive, most simulation studies today are implemented using a simulation package. As indicated in [D-7], these packages provide the end user with reduced programming requirements; natural framework for simulation modelling, conceptual guidance, automated gathering of statistics, graphic symbolism for communication, animation and flexibility to change the model.

D.3 WITNESS Simulation Software

The following sections provide information on Witness simulation tool used in this application.

D.3.1 Witness Advance features:

As described earlier, witness provides techniques allowing rapid construction of complex and user friendly models. These features are described as follows:

D.3.1.1 Designer Element Library:

The designer elements are a palette of user-defined elements with display and detail parameters already defined. These building blocks are stored in a file called STARTUP.MOD, which is opened when Witness is initiated. The designer elements are used to generate the basic elements included in the model. These elements can be selected and dropped into the main modelling window. Using the Display and Detail forms on the element, the user can further customise the elements.

D.3.1.2 Sub-Models:

These are the pieces of other models that the modeller might need to use over. The user can save a model as a sub-model (.sub file) from the menu bar. This allows the selection of the elements to be included in the sub-model. The sub model can be accessed from other models by selecting and opening the appropriate .sub file. This feature reduces model development time as the user does not need to rewrite some existing functionalities of interest.

D.3.1.3 Module and Hierarchical Modelling:

Module is a designer element that allows one to encapsulate a sub-model into it. Once a sub-model is assigned to a module element, the module can then enter the designer element window and then dragged and dropped into the model development window. This feature allows hierarchical modelling, which is a principle of object oriented programming approach.

D.3.1.4 Object linking and Embedding-OLEII:

Witness is capable of acting as an OLEII server. This allows running Witness from an OLEII controller application such as Microsoft Excel, Access and Visual Basic applications. Also, Witness provides a Witness Command Language (WLC), using this command one can run witness automatically by setting this command in a special file called a command input file (.wcl file). This facility is especially useful for running a series of experiments where long batch runs are punctuated by a few commands that set up the next experiment. This feature saves model building time to use this data directly. Also using this feature, witness can be integrated with other OLEII applications to exchange information and therefore preventing direct interface with the end users, who may have no prior knowledge on simulation modelling.

D.3.1.5 Communication Features

To communicate the generated results form simulation model more effectively. Witness represents a number of capabilities as follows:

- **Reporting:** The witness provides built-in reports, which are generated automatically and presents results in a standard grid format. Also, a variety of charts and graphs are available for further customisation.
- **Process view:** this provides a flow-charted view of the model in addition to the more typical simulation model view. This type of view provides a better client communication.
- **AVI File:** Witness allows recording both the virtual reality and 2-D simulation runs to a windows video (.avi) file format. These files may then be distributed for presentation purposes.

References D

D - 1. Carson, J.S., **Introduction to modelling and simulation.** Proceeding of the 2003 Winter Simulation Conference, S. Chick, P.J. Sanchez, D.Ferrin, and D.J. Morrice, Eds., (2003), PP:7-13.

D - 2. Thesen, A., Travis,L.E., **Introduction to simulation.** Proceeding of the 1990 Winter Simulation Conference, Osman Balci, Randall P.Sadowski, Richard E. Nance (eds.), (1990), PP: 14-21.

D - 3. Banks, J., Carson, J.S., **Introduction to discrete-event simulation.** Proceeding of the 1986 Winter Simulation Conference, J.Wilson, J.Henriksen, S. Roberts (eds.), (1986), PP: 17-23.

D - 4. Law, A.M., Kelton, W.D., ***Simulation modelling and analysis.*** 3rd ed. McGraw-Hill Book Co, (2000), Singapore, ISBN: 0-07-116537-1.

D - 5. Nance, R.E., **The time and state relationships in simulation modelling.** Communications of the ACM, Vol. 24, (1981), PP:173-9.

D - 6. Pritsker, A.A.B., Pegden, C.D., **Introduction to simulation and SLAM.** Halstead Press, (1979).

D - 7. Anu, M., **Introduction to modelling and simulation.** Proceeding of the 1997 Winter Simulation Conference, Eds, Andradottir, K.J. Healy, D.H. Withers, and B.L. Nelson, (1997), PP: 7-13.

Appendix E: LPG Supply Chain

E.1 Activity in supply-chain management

Some key management activities within supply chain include:

- **Site selection**: this involves how to locate a facility to achieve the outmost response. This involves not only the manufacture's facility but also the supplier's factory, warehouse and distribution centres.
- **Forecasting:** this involves forecasting demand for customers, which is the activity that sets into motion planning and the material flow in the supply chain.
- **Development of an operations plan:** this activity corresponds to the sales needs. This includes all the integrated activities, such as development of the master production schedule, the material requirements plan and the operations schedule.
- **Raw Material Management:** this involves, raw material inventories, work in progress and finished goods, such that there is sufficient materials and not to have stock out situations, but not too much so that costs are unnecessarily high.
- **Purchasing:** this activity ensures that the right materials of the expected quality are delivered at the right location on the specific date.
- **Distribution requirements:** this activity involves route planning, and transportation for finished products.

E.1.1 Site Selection:

In the supply chain, the location of the operating facility is a critical factor in timely delivery of products and services. The site selection or facility location for either manufacturing plants or distribution centres is important in order to reach new markets, increase production capacity and/or serve clients better. There are many factors that were considered in selecting sites for the LPG bottling plants and distribution centres across the country such as staffing, local conditions, labour, local population, proximity of sub contractors and suppliers and availability of energy and raw materials. Furthermore, as transportation cost adds significantly to the cost of finished products, the availability of transportation facilities and networks are important for the delivery of raw materials and dispatching finished goods.

E.1.2 Forecasting:

Forecasting, or estimating, the demand for the LPG cylinders is the starting point for all operating activities. It is the trigger that sets the supply chain in motion and it initiates activities such as production plans, personnel needs, capacity levels of equipments, purchasing raw materials and transportation requirements.

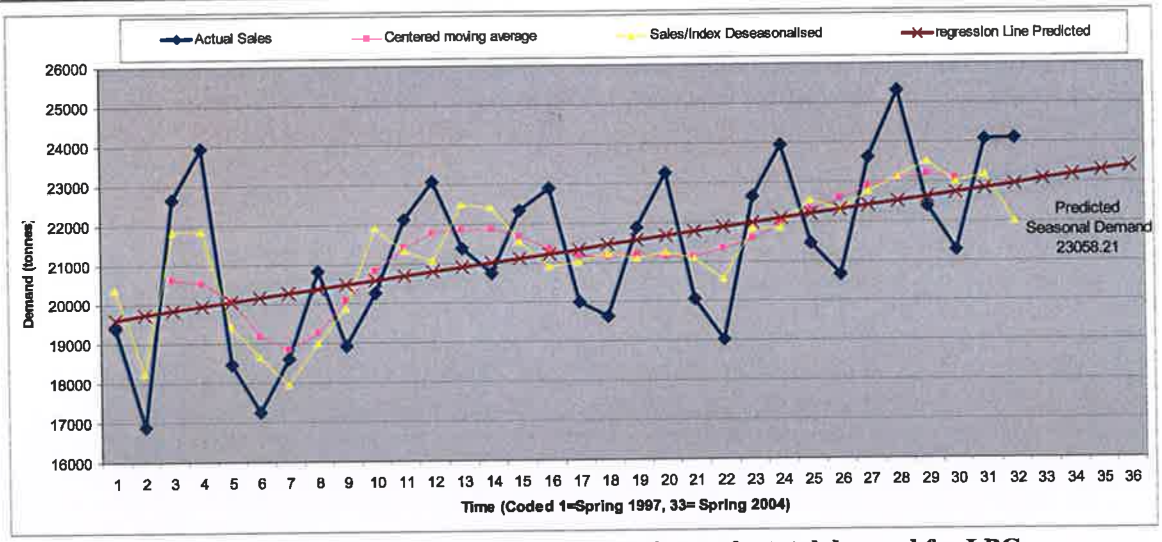


Figure E-58 the forcasting model used to estimate the total demand for LPG

Forecasting the demands for LPG is performed at both bottling plants and logistics departments. Figure E-58, represents a time series forecasting method which allows the managers to estimate the LPG demand in future based on historical data available from past activities.

As it is evident from Figure E-58, the demand for LPG shows an increasing trend. Also, for any given year, the demand for LPG peaks during the autumn and winter (e.g. code 3 & 4) and falls at spring and summer time (e.g. code 1 & 2). Using the established regression line from this chart, the managers could predict the future demand for spring 2004 as shown in this figure.

In addition, bottling plant managers could predict the upcoming seasonal demands for LPG using the historical data. The time horizon usually used by the production managers in bottling plants covers a short range, which involves a time span from a few weeks up to three months. Figure E-59, illustrates a typical forecasting chart used by the production manager for predicting seasonal demands for LPG.

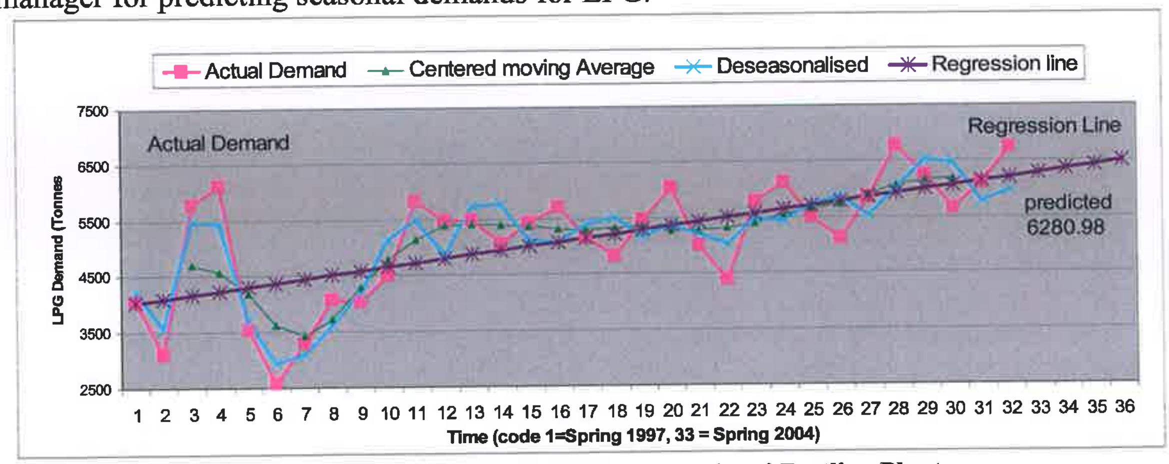


Figure E-59 a forcasting model used for Karaj Bottling Plant

The logistics department could also make use of the forecasted demands from bottling plants to set schedules and delivery plans to manage purchasing and transportation of the required LPG from refineries to the required destinations.

E.1.3 Development of an operations plan

The objective of the operation plan is to specify all the activities necessary in order to produce end products, or to provide the required service for a customer in timely manner. The operation plan is generated based on customer orders. An operation plan for LPG processing and distribution could be considered in short and medium range plans.

The short range may include activities such as scheduling LPG processing programs, personnel job assignments, organising deliveries of raw materials and shipment of cylinders. These plans are usually developed for a month period. The medium range plans my include activities such as selection of new subcontractors, installation of new machines and equipments and production plans.

As Shown in Figure E-60, the operation plan is driven by customers' demands or by forecast demands from each bottling plant located across the country. The production managers specify the demands on monthly basis. The Logistics manager in the central office is responsible to establish an aggregated plan. The aggregate plan is the development of an estimate for future needs of LPG cylinders. This estimate is usually generated from inputs provided by bottling plants manager concerning orders from clients anticipated or forecasted orders. The estimate is established by aggregating the LPG needed by each plant into a total demand. The logistics manager uses the total demand to decide how to use resources such as labour, vehicles and machines to meet the forecast/ actual demand in a cost effective manner within the constraints and capacity of the facility.

Based on this plan, the logistics manger establishes the overall monthly raw material (i.e. Butane gas) requirements plan and the transportation schedule for the company. The raw material plan indicates the materials and their respective quantity to be ordered from outsides considering the current inventory levels. The transportation schedule indicates the time and quantity of raw material to be transferred from refineries to each Bottling plant in the system. The bottling plant manager would use the transportation schedule to approximate the raw material arrivals into the bottling plant to establish the operation schedule for their respective plant.

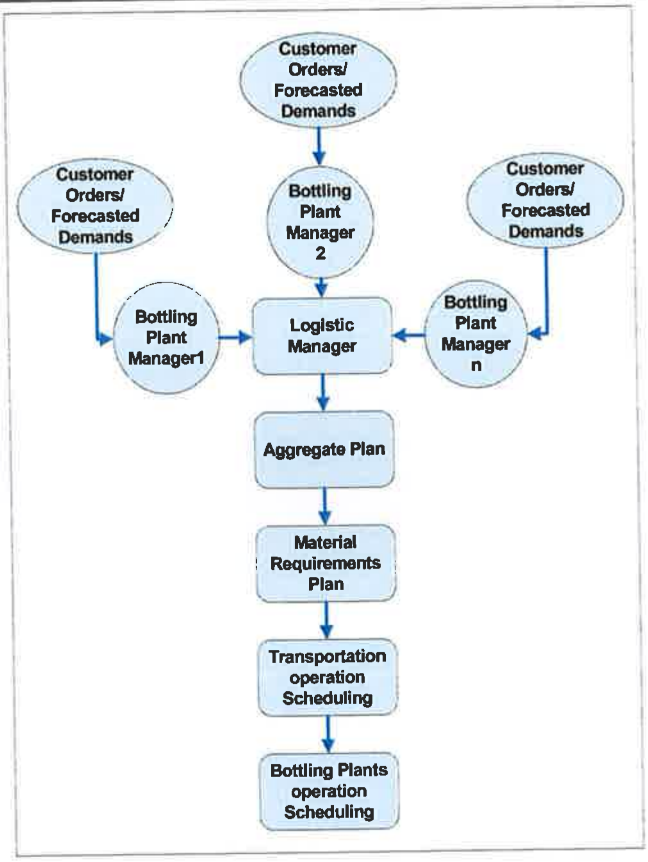


Figure E-60 Planning stages in supply chain operations

E.1.4 Raw Material Management:

In the supply chain one of the key issues that have to be managed is inventory. The inventory covers a vast spectrum of material that is being transferred, stored, consumed and produced during the considered organisation normal course of activities. In general there are two approaches in inventory management. One is to keep the inventory as low as possible in the supply chain, thus free up some assets for other activities. Alternatively, move the inventory in the supply chain as fast as possible for delivery to the customer to obtain the gains in the value added. The inventory exists in this supply chain could be defined into the following categories:

- **Raw Materials**: This refers to the Butane gas produced in the refineries. These are the starting elements for the gas cylinder filling process. These materials are purchased from the refineries and they are transferred and stored in the bottling plants for processing.
- **Transit Inventory**: This category refers to those materials purchased from refineries and they are being transported to the bottling plants.
- **Work in Progress:** This represents the parts that are moving through the production operation. At each stage, the pieces are being modified and value is being added. All material between raw materials and finished product in each bottling plant are considered as work in progress inventory.
- **Finished products:** These are the gas cylinders or bulk products that are stored in the dispatching inventories. The end user may use these products or they may be further destined for another industrial user.

The logistics department is responsible to monitor inventory levels in the supply chain to prevent any breakdowns in the system. For this reason, this department checks the available inventories with each supplier on daily basis. This is done in order to make sure that there is enough supply to be taken for the required consumption.

In addition the inventories at the bottling plants are monitored every day. This includes the raw material available in the reservoirs and the finished goods stored in the dispatching inventories.

To monitor the inventories at each bottling plant, the department uses an in house inventory control chart sheets. This control chart includes upper & lower control limits, which are established to prevent any stock starvations or over supply of the inventories. These charts are established on monthly bases. To establish these charts, the historical data on the daily demand for the last five years for the considering period is collected. The mean (μ) and the standard deviations (σ) for daily demands are calculated based on the collected data.

The safety stock level (SS) is calculated based on the company service level policy, which is set at 98%. To calculate the safety stock level, Z value is obtained from the normal distribution table. The Z value is the number of standard deviations the demand is from the mean value. The following shows how the safety stock level is calculated:

$$SS = Z\sigma$$

The Lower bound on the control chart is established as follows:

$$L.b. = \mu + SS$$

The upper bound on the inventory control chart is established based on the safety regulations. This regulation suggests that 10% of the LPG reservoir's capacity is not to be used during the warm seasons. Therefore, the upper bond for summer times is calculated as follows, otherwise it stays the same as the actual capacity:

U.b. = actual capacity of reservoirs- (10% * actual capacity of reservoirs)

The following chart shows a control chart, which was established for the month of August 2004 for one of the bottling plants.

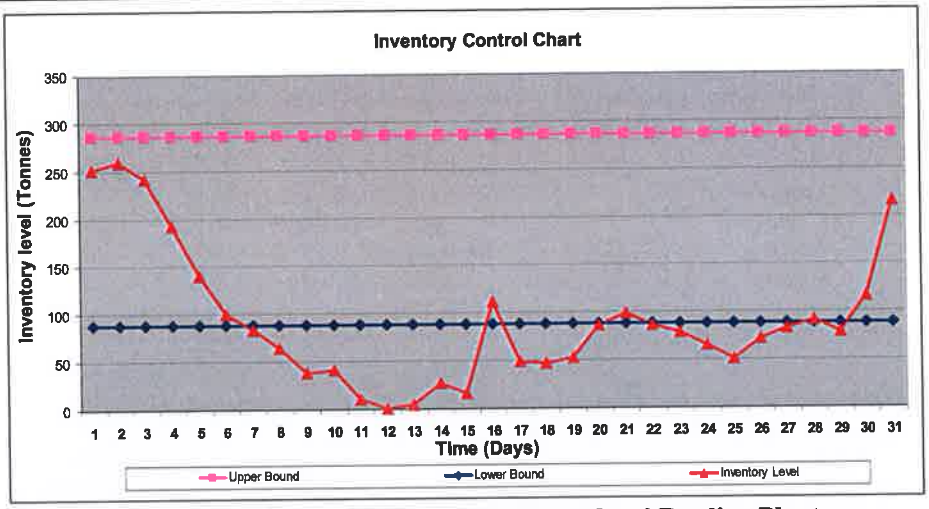


Figure E-61 Inventory control chart for Karaj Bottling Plant

E.1.5 Purchasing

In general, Purchasing is buying of materials and services from outside sources and it involves transferring materials from sources to where it is needed. Purchasing the Butane gas from the refineries is one of the main activities in this supply chain and it is the start of the supply chain. This purchasing activity is initiated based on the clients demand for gas cylinders.

Bottling plants manager based on the clients' requirements establishes production plan. Based on this plan the production manager decides what needs to be purchased. The logistics manager mainly does the purchasing. Therefore, to make sure that there is no break in this supply chain there is a need of close cooperation between bottling plant managers, production personnel and the logistics manager.

In this organisation, the purchasing activity concerning the raw material is centralised. The centralised purchasing means that all purchased items for every division are processed through one department, which in this case is the logistics department. The centralised purchasing could have a number of advantages such as:

- Reduces administration costs, the time taken to negotiate orders, billing costs and consequently the overall costs.
- Relation with suppliers is simplified, as there would be fewer interlocutors.

E.1.6 Steps in purchasing process:

Figure E-62, illustrates the overall steps involved in purchasing Butane gas for bottling plants in the supply chain. These steps are further described as follows:

- **Purchase requisition:** Production managers from bottling plants in the chain issue a purchase or martial requisition. These requisitions are sent to purchasing in the logistics department. This document is a request to buy Butane gas for consumption by bottling plants. It includes the quantity to purchase, the delivery date, the account details to charge and the location to deliver the order.

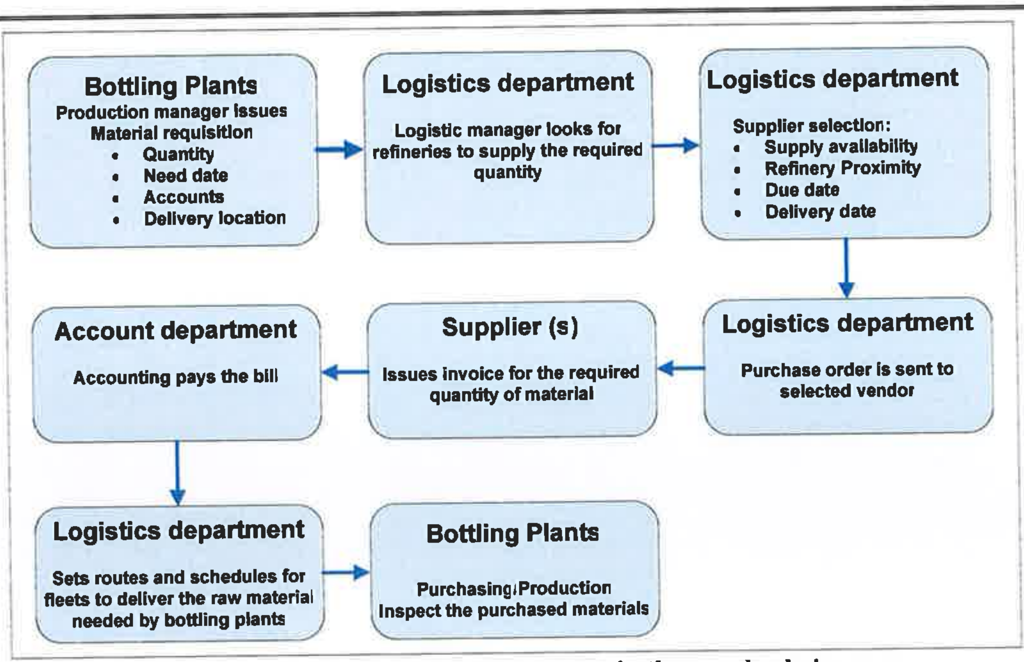


Figure E-62 the purchasing process in the supply chain

- **Request for supply:** the purchasing in the logistics department would look for different refineries that could provide the requested demand.
- **Supplier Selection:** the best supplier is selected based on the following factors considered by the logistics manager: How far the refineries are located from the demand points, How much gas supply could the refineries provide, Could the available resources such as the vehicles and manpower mange the transportation of the required raw material. Would the transportation from the selected refineries affect the agreed delivery and due dates.
- **Purchase order:** the purchase department would send out a purchase order to the selected Refinery.
- **Supplier invoice:** the selected supplier provides the purchase department with an invoice.
- **Payment:** based on the terms and conditions offered by the suppliers and agreed by the supplier manager, the payment is made by the account department.
- **Vehicle routing and scheduling:** the logistics manager sets a routing and scheduling plan for transporting the required row materials to the respective bottling plants.
- **Inspection:** upon arrival of the raw materials by the ordered bottling plant, the shipment is inspected and the material is received. Also, the logistics department is updated with this shipment.

E.2 Supply chain Costs:

Figure E-63, gives a break down of the LPG supply chain costs according to various steps exists in the chain from the supply of the raw material to the delivery of the finished products. Based on the cost analysis performed in this supply chain, 68% of the total supply chain cost is related to the upstream operations. Also, 10% of the cost is related to the transformation process and 22 % to the down stream activities. The biggest cost in this supply chain is related to the cost of LPG transportation from the refineries to the bottling plants. The downstream transportation is subcontracted to reduce the overall transportation cost, which is reduced to 19% of the total cost.

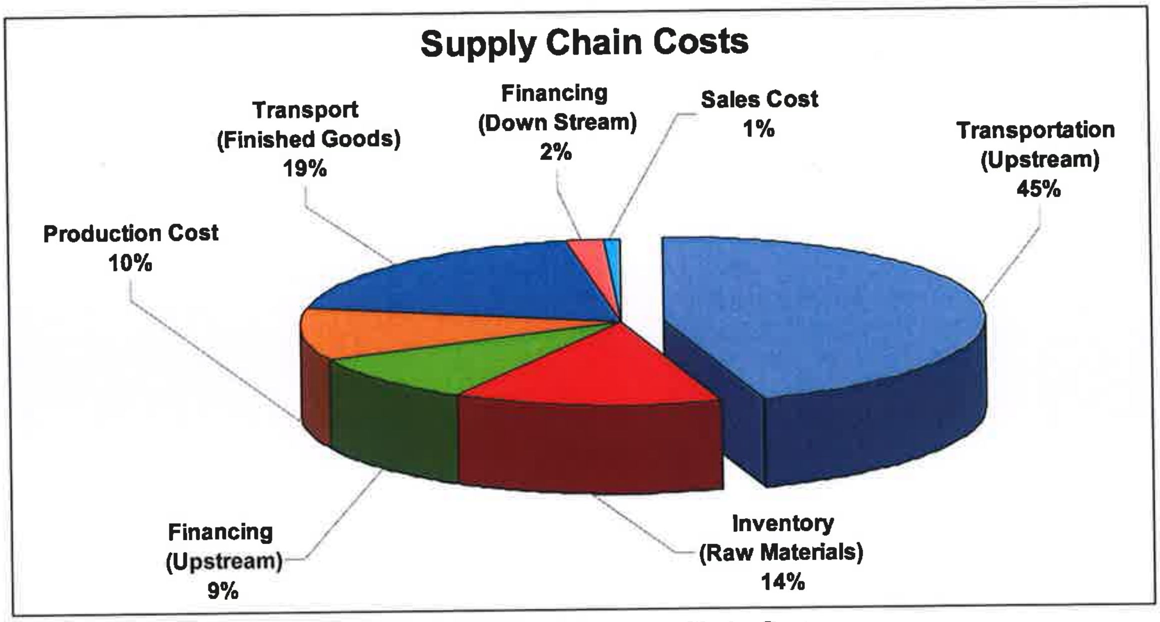


Figure E-63 LPG Supply Chain Cost

Appendix F: Data collected for Simulation Model Development

F.1 Data collected for monthly capacities provided by refineries:

The following tables indicate the monthly supply capacities provided by different refineries in the supply chain.

F.1.1 Abadan:

Abadan Refinery												
	March	April	May	Jun	July	Agust	September	October	November	December	January	February
1	367	367	367	367	367	367	200	200	367	367	367	367
2	225	225	225	225	325	367	225	367	367	250	300	300
3	225	225	225	225	250	600	225	225	225	200	200	225
4	225	225	225	225	250	225	345	345	225	225	225	489
5	195	195	195	195	300	0	383	225	225	225	225	225
6	500	300	300	300	300	150	195	275	0	195	300	195
7	367	367	367	367	367	367	367	367	367	367	367	300
8	333	273	349	299	319	357	173	148	333	296	288	339
9	135	187	130	198	263	330	192	305	287	221	261	297
10	131	184	172	138	172	541	221	181	179	150	178	140
11	175	171	208	152	203	195	277	324	160	147	209	421
12	144	169	112	167	227	-95	330	140	129	223	183	151
13	475	211	245	281	280	139	114	252	-88	112	233	132
14	325	280	366	290	296	343	322	337	276	342	365	301
15	362	266	343	301	224	357	55	99	236	291	247	359
16	189	216	134	180	311	352	105	261	337	123	175	237
17	181	79	116	166	164	529	104	201	206	72	59	183
18	99	132	165	206	243	140	339	238	131	165	110	383
19	105	89	78	108	175	0	268	157	110	130	80	115
20	435	209	269	240	270	19	143	237	0	104	200	182
21	245	308	324	266	227	360	308	296	301	328	345	301
22	359	339	278	240	178	342	21	100	202	224	248	173
23	82	153	50	141	138	341	79	334	184	177	195	194
24	196	28	158	67	83	594	183	107	41	65	180	74
25	137	48	194	103	133	205	273	256	221	162	168	442
26	96	160	175	148	128	0	230	80	86	172	158	42
27	374	234	269	186	149	25	6	159	0	92	135	149
28	278	194	348	170	204	322	203	290	181	284	361	217

F.1.2 Arak:

Arak Refinery												
	March	April	May	Jun	July	Agust	September	October	November	December	January	February
1	800	600	600	600	600	600	604	590	600	600	600	600
2	724	636	487	1106	903	1190	800	800	800	800	800	800
3	1200	1200	1000	1000	1000	600	1000	1000	600	1070	1205	735
4	550	600	600	1802	1400	1200	1200	1200	1200	1200	1200	1200
5	540	510	510	596	596	540	770	600	550	550	550	550
6	550	540	240	550	600	600	500	500	520	520	540	540
7	486	508	450	700	350	500	300	300	400	400	400	400
8	761	506	526	572	581	542	528	529	539	564	591	517
9	675	550	437	1059	808	1121	772	733	743	780	758	795
10	1109	1163	924	904	968	568	921	937	553	1004	1131	688
11	502	586	576	1792	1397	1138	1187	1189	1171	1148	1103	1137
12	451	501	413	570	537	468	720	509	472	498	494	526
13	537	458	187	531	598	541	417	421	430	489	535	477
14	464	495	446	617	312	433	237	293	347	347	399	385
15	697	549	581	522	564	465	561	480	481	510	494	601
16	606	620	424	1060	758	1153	736	783	750	695	713	778
17	1083	1143	873	965	930	494	900	928	589	1056	1059	656
18	539	480	575	1723	1278	1094	1055	1097	1098	1175	1120	1175
19	472	387	396	503	542	426	634	593	408	543	424	453
20	509	424	187	431	549	536	354	466	461	457	433	503
21	482	469	316	661	289	477	250	158	365	299	371	388
22	746	515	563	468	521	573	587	591	567	597	545	505
23	644	545	339	920	828	1157	658	625	700	667	618	722
24	1021	1020	836	867	958	522	851	929	537	1035	1173	549
25	488	504	438	1691	1261	1008	1033	1132	1059	1076	1056	1104
26	445	485	478	455	556	488	684	535	388	354	494	374
27	538	446	66	520	497	540	340	455	322	495	418	495
28	315	444	445	595	308	365	193	129	337	218	281	267

F.1.3 Bandar Abas:

BandarAbas Refinery												
	March	April	May	Jun	July	Agust	September	October	November	December	January	February
1	1666	2074	2191	859	1047	1053	1654	2159	1413	1977	1973	1842
2	1458	1637	1030	1503	1592	1478	1759	1728	1955	2117	2137	1819
3	1456	1670	1567	1324	1347	2163	2176	1575	1976	1450	1820	1839
4	1132	1383	1386	1407	1318	1390	1367	1348	1413	1488	1491	1463
5	1154	931	1022	490	485	1017	1094	1272	1272	1736	1136	1330
6	1800	1800	1800	1600	1500	1500	1177	1154	1196	1850	1200	1154
7	650	600	550	882	1200	1631	1600	2000	1272	2000	1800	1800
8	1590	2040	2101	839	1003	1032	1634	2139	1389	1963	1943	1786
9	1367	1637	1019	1425	1506	1453	1721	1645	1956	2088	2085	1761
10	1418	1666	1556	1305	1321	2080	2082	1526	1971	1402	1737	1782
11	1098	1367	1356	1335	1263	1365	1335	1334	1374	1432	1470	1391
12	1064	903	925	487	410	964	1039	1178	1180	1716	1084	1268
13	1734	1748	1765	1511	1416	1447	1103	1108	1101	1798	1178	1071
14	589	524	471	809	1173	1601	1562	1935	1272	1945	1741	1795
15	1630	1955	2174	779	1046	918	1592	2071	1331	1946	1929	1708
16	1311	1534	943	1463	1572	1408	1632	1618	1893	1981	2035	1680
17	1386	1635	1535	1195	1316	2143	2151	1530	1961	1325	1754	1760
18	1092	1256	1348	1353	1265	1339	1351	1255	1325	1450	1354	1364
19	1007	884	1011	408	419	980	1016	1252	1160	1598	1069	1289
20	1778	1781	1744	1579	1443	1423	1094	1117	1146	1710	1065	1042
21	503	597	521	827	1139	1602	1592	1930	1211	1939	1729	1754
22	1524	1877	2172	782	978	907	1458	2129	1368	1837	1895	1712
23	1375	1505	896	1432	1515	1304	1674	1542	1830	1933	2086	1799
24	1266	1624	1457	1238	1152	2143	2004	1385	1818	1293	1731	1753
25	1104	1322	1377	1348	1309	1241	1270	1172	1250	1454	1374	1269
26	1028	876	970	371	474	850	940	1117	1146	1621	1045	1162
27	1793	1782	1692	1533	1416	1426	1089	1030	1169	1762	1117	1112
28	553	531	524	852	1158	1590	1546	1947	1261	1871	1706	1793

F.1.4 Esfahan:

Esfahan Refinery												
	March	April	May	Jun	July	Agust	September	October	November	December	January	February
1	3884	2921	2759	2434	2305	2325	2758	2576	2492	3720	3644	3396
2	3300	3300	3300	3300	2900	3000	2921	3054	3086	3516	3496	3229
3	3972	3472	3775	3295	2943	4248	3518	3475	3922	3300	3300	3300
4	4978	4617	4554	3000	3019	3593	3285	3264	3570	3686	4025	3773
5	4586	5093	5002	4724	4724	4908	3919	4378	4441	4633	5064	5001
6	3600	3600	3600	3600	3700	.3400	4071	3942	4449	4449	4619	4586
7	4504	4330	3720	4399	3600	3640	4050	3900	4150	4100	4150	3680
8	3852	2856	2753	2339	2228	2264	2706.92	2518	2406	3647	3628	3345
9	3248	3227	3207	3224	2826	2956	2875.14	3045	2995	3495	3417	3193
10	3969	3403	3709	3254	2858	4183	3496.52	3399	3858	3208	3253	3259
11	4976	4599	4555	2972	2982	3581	3250.08	3203	3505	3596	3972	3682
12	4503	5065	4950	4679	4700	4885	3919.21	4355	4422	4551	5013	4967
13	3537	3588	3547	3512	3635	3318	4045.42	3860	4372	4368	4591	4586
14	4447	4326	3721	4264	3512	3510	4006.52	3799	4013	3970	4071	3623
15	3738	2804	2707	2426	2259	2269	2737.91	2451	2484	3702	3544	3305
16	3239	3169	3290	3202	2867	2986	2813.69	2973	3033	3501	3350	3176
17	3907	3430	3661	3206	2927	4151	3432.38	3473	3836	3283	3284	3157
18	4899	4532	4416	2995	2905	3553	3275.01	3153	3423	3583	3915	3739
19	4577	5064	4922	4695	4591	4772	3852.72	4339	4416	4613	4955	4949
20	3470	3583	3528	3501	3637	3330	4048.09	3820	4329	4431	4518	4522
21	4424	4323	3676	4360	3549	3515	3953.62	3859	4084	4047	4077	3605
22	3835	2851	2564	2382	2215	2143	2691.75	2483	2340	3590	3476	3227
23	3283	3204	3300	3211	2890	2855	2779.12	3012	3003	3327	3492	3130
24	3970	3383	3674	3114	2822	4246	3506.28	3369	3731	3261	3150	3208
25	4931	4465	4411	2993	2827	3482	3129.48	3103	3477	3510	3951	3742
26	4488	5009	4904	4645	4621	4757	3821.41	4339	4268	4452	5049	4913
27	3469	3515	3513	3444	3606	3358	3990.22	3874	4432	4369	4570	4413
28	4415	4216	3682	4315	3456	3451	3888.17	3825	4103	3939	4093	3637

F.1.5 Tehran:

Tehran Refinery												
	March	April	May	Jun	July	Agust	September	October	November	December	January	February
1	600	600	350	350	200	200	200	350	600	600	600	600
2	600	600	500	400	500	500	600	600	600	600	600	600
3	600	600	600	600	600	600	550	550	400	500	500	600
4	400	500	550	550	250	250	450	500	500	500	500	400
5	500	471	471	350	350	535	450	500	500	400	400	400
6	750	780	770	750	800	700	500	550	580	580	570	500
7	680	538	500	450	540	500	700	700	900	900	900	750
8	557	508	339	346	154	160	166	310	507	537	587	549
9	597	538	493	310	467	494	593	550	589	555	526	542
10	509	536	577	575	564	528	458	481	361	474	427	514
11	336	466	471	531	212	173	373	471	459	430	433	321
12	479	426	404	269	263	465	451	466	422	341	377	397
13	673	695	770	704	719	656	475	483	508	524	504	487
14	677	537	477	405	534	413	688	662	899	827	823	666
15	467	522	340	340	141	52	157	301	601	524	575	514
16	556	488	464	259	389	440	537	571	585	550	531	556
17	576	495	588	587	569	485	529	478	267	360	435	562
18	340	474	545	489	104	211	419	376	434	481	403	278
19	388	380	420	223	264	396	413	403	395	274	304	357
20	643	707	656	653	662	587	498	409	446	551	485	382
21	615	485	431	348	396	478	650	615	854	836	799	704
22	452	493	247	162	148	132	154	247	472	591	586	423
23	476	502	306	379	454	483	494	585	542	572	486	448
24	462	453	531	515	594	567	536	545	270	377	440	597
25	361	331	376	546	69	152	325	443	399	316	302	351
26	387	352	369	241	236	493	320	343	392	263	229	254
27	645	716	755	717	783	669	383	403	518	427	548	405
28	656	464	418	297	399	454	542	570	790	841	860	652

F.2 Data collected for waiting times at Refineries

F.2.1 Abadan waiting time

Refineries												
Abadan	January		February		March		April		May		Jun	
Year	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2004	495	1146	483	1080	210	864	240	642	298	863	220	1200
2003	390	1440	499	871	393	741	376	586	231	882	185	1264
2002	432	1253	327	816	399	900	284	631	303	896	213	1215
2001	364	1029	357	845	225	896	336	714	310	924	211	1235
2000	358	1348	392	825	252	720	345	598	253	897	184	1298
1999	522	1122	348	923	257	894	309	686	281	846	205	1204
1998	373	1316	322	874	275	834	366	554	253	917	217	1255
1997	481	1413	428	835	215	723	317	720	308	869	212	1261
1996	436	1053	240	905	359	862	359	693	301	919	209	1199
1995	300	1412	443	1006	233	725	360	612	288	881	216	1257
Min/Max	300	1440	240	1080	210	900	240	720	231	924	184	1298
Refineries												
Abadan	July		Agust		September		October		November		December	
Year	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2004	252	1119	188	884	188	907	390	1250	169	935	210	1032
2003	291	1114	208	901	183	917	342	1182	297	904	201	1132
2002	265	994	274	894	189	922	381	1440	197	890	211	1125
2001	230	1023	277	914	219	950	385	1274	309	920	200	920
2000	238	1028	304	925	148	897	323	1166	247	895	222	1119
1999	196	1169	257	911	203	895	345	1294	251	967	207	996
1998	204	1200	235	919	198	923	407	1315	224	905	212	1007
1997	186	1068	204	891	183	902	348	1237	279	887	171	907
1996	225	1157	133	907	177	915	386	1384	242	899	205	1000
1995	227	1116	306	886	211	898	380	1367	230	971	187	917
Min/Max	186	1200	133	925	148	950	323	1440	169	971	171	1132

F.2.2 Arak waiting time

Refineries												
Arak	Junuary		February		March		April		May		Jun	
Year	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2004	401	962	378	965	337	1354	274	1006	374	1079	366	1303
2003	357	989	339	1029	396	1373	262	1010	296	1067	296	1317
2002	404	1025	416	1065	374	1348	241	1036	377	1118	288	1297
2001	365	976	359	968	391	1356	289	1039	385	1057	313	1281
2000	388	974	430	999	322	1296	228	970	335	1071	386	1296
1999	324	963	427	1010	377	1344	203	1063	304	1107	387	1229
1998	381	1036	351	996	344	1286	251	1025	380	1068	386	1217
1997	352	1002	421	965	392	1317	232	1047	380	1077	321	1241
1996	410	1061	350	1059	391	1372	249	1067	300	1132	322	1288
1995	375	998	346	1039	387	1302	213	978	305	1142	335	1219
Min/Max	324	1061	339	1065	322	1373	203	1067	296	1142	288	1317
Refineries	July		Agust		September		October		November		December	
Arak	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Year	192	690	190	810	204	989	211	739	144	724	146	925
2004	244	731	199	855	153	987	307	759	200	755	148	887
2003	173	725	166	853	202	910	216	787	171	766	210	891
2002	257	646	231	850	181	931	300	782	186	780	121	902
2001	182	660	232	909	221	978	251	748	226	751	140	979
2000	215	717	265	863	225	937	241	754	165	724	185	959
1999	243	649	254	878	247	1000	245	724	234	781	149	982
1998	261	683	203	817	194	936	306	760	158	720	216	894
1997	165	638	235	824	196	919	293	692	220	681	208	962
1996	189	693	231	910	181	916	219	745	243	755	172	885
1995	165	731	166	910	153	1000	211	787	144	781	121	982
Min/Max	165	1061	339	1065	322	1373	203	1067	296	1142	288	1317

F.2.3 Bandar Abas waiting time

Refineries												
Bandar Abas	Junuary		February		March		April		May		Jun	
Year	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2004	349	1271	347	1206	291	1255	183	300	203	846	210	1205
2003	404	1291	380	1286	369	1276	157	263	227	901	219	1140
2002	398	1240	366	1259	373	1193	140	222	185	801	292	1158
2001	424	1313	353	1287	355	1216	194	257	275	851	254	1116
2000	327	1253	311	1230	381	1233	164	237	201	873	215	1154
1999	401	1284	398	1244	295	1237	188	298	270	854	274	1172
1998	382	1267	406	1236	297	1228	155	265	277	874	294	1156
1997	388	1332	382	1203	391	1261	223	212	195	836	238	1124
1996	361	1288	332	1287	374	1254	182	242	225	826	219	1114
1995	360	1232	376	1290	306	1231	231	272	214	857	258	1174
Min/Max	327	1332	311	1290	291	1276	140	300	185	901	210	1205
Bandar Abas	July		Agust		September		October		November		December	
Year	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2004	147	954	190	770	215	811	224	501	158	644	239	951
2003	153	947	183	794	240	760	242	532	199	626	267	1024
2002	162	990	218	711	214	716	283	567	132	680	247	1035
2001	178	1012	165	786	235	771	278	486	164	651	246	989
2000	164	990	178	707	173	747	229	520	121	600	306	1000
1999	224	931	205	741	150	792	212	568	163	586	251	1004
1998	188	914	234	753	204	769	263	544	202	601	206	949
1997	179	950	219	752	163	733	256	560	179	672	256	972
1996	125	993	142	799	225	732	231	473	161	600	225	956
1995	134	916	204	739	206	781	244	473	137	592	301	993
Min/Max	125	1012	142	799	150	811	212	568	121	680	206	1035

F.2.4 Esfahan waiting time

Refineries												
Tehran	January		February		March		April		May		Jun	
Year	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2004	351	996	370	932	245	949	227	917	327	866	233	1275
2003	355	918	422	1022	295	1005	282	908	275	867	222	1227
2002	334	961	402	1005	243	934	296	921	307	890	276	1238
2001	410	932	328	977	251	1011	206	919	310	852	307	1242
2000	417	962	425	954	294	970	245	994	284	847	261	1291
1999	377	918	330	1008	334	1011	224	905	250	924	308	1199
1998	356	1013	418	960	289	944	243	895	266	871	267	1239
1997	386	941	379	1019	343	926	296	964	305	902	290	1195
1996	411	1000	356	1019	316	923	289	974	312	834	225	1288
1995	370	918	390	968	335	1005	246	923	334	847	260	1204
Min/Max	334	1013	328	1022	243	1011	206	994	250	924	222	1291
Tehran	July		Agust		September		October		November		December	
Year	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2004	187	696	257	821	211	594	191	806	232	897	240	597
2003	208	701	224	765	185	597	147	885	191	918	213	571
2002	233	662	227	826	175	579	218	877	197	865	248	601
2001	190	641	174	821	130	645	176	852	196	879	182	609
2000	179	645	252	732	153	631	191	842	211	957	150	641
1999	195	708	161	773	160	551	223	864	219	948	203	604
1998	180	690	187	776	191	600	142	852	253	904	169	594
1997	201	632	170	818	156	640	152	837	217	924	214	638
1996	141	628	219	760	196	648	169	866	282	870	211	658
1995	194	666	189	815	173	555	141	805	237	910	172	585
Min/Max	141	708	161	826	130	648	141	885	191	957	150	658

F.2.5 Tehran waiting time

Refineries												
Esfahan	January		February		March		April		May		Jun	
Year	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2004	372	1349	294	1010	272	1286	223	1138	329	1072	277	956
2003	358	1362	363	974	327	1274	255	1191	312	1118	295	975
2002	369	1440	279	951	304	1278	242	1111	272	1093	214	947
2001	355	1361	286	942	350	1263	180	1138	308	1086	217	948
2000	300	1410	266	938	360	1213	279	1140	350	1059	289	996
1999	380	1413	353	997	371	1210	260	1101	337	1098	197	1047
1998	308	1366	313	963	347	1220	192	1181	356	1089	227	961
1997	399	1439	315	1026	313	1240	255	1161	317	1118	284	1009
1996	369	1340	327	938	359	1251	253	1131	371	1082	222	966
1995	399	1383	364	988	315	1229	264	1200	363	1090	265	994
Min/Max	300	1440	266	1026	272	1286	180	1200	272	1118	197	1047
Esfahan	July		Agust		September		October		November		December	
Year	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
2004	182	791	208	967	191	511	251	935	144	1151	236	617
2003	179	835	244	961	163	540	273	906	169	1111	174	665
2002	157	868	172	1045	214	440	287	950	155	1178	159	694
2001	217	821	219	996	165	452	297	949	126	1164	252	657
2000	162	800	177	981	138	503	287	977	207	1153	173	647
1999	150	819	213	1033	163	537	265	963	154	1081	235	697
1998	213	828	201	1010	190	473	248	1006	124	1163	235	636
1997	191	810	192	968	202	536	234	978	211	1079	256	650
1996	222	775	217	970	235	459	272	949	199	1092	233	618
1995	233	861	197	1028	190	523	275	925	196	1155	223	632
Min/Max	150	868	172	1045	138	540	234	1006	124	1178	159	697

F.3 Data collected for loading times at Refineries

	Year	Month	Average Loading Time (Minutes)				Tehran
			Abadab	Arak	Bandar Abas	Esfahan	
1	2002	1	120	73	207	133	210
2	2002	2	164	180	210	120	212
3	2002	3	205	186	210	124	216
4	2002	4	208	190	300	119	246
5	2002	5	210	169	208	118	210
6	2002	6	298	152	211	120	212
7	2002	7	213	185	146	125	205
8	2002	8	215	180	212	117	206
9	2002	9	214	179	211	125	208
10	2002	10	209	180	252	120	138
11	2002	11	212	179	209	123	210
12	2002	12	126	176	212	142	214
13	2003	1	159	194	205	119	211
14	2003	2	210	180	215	117	214
15	2003	3	213	184	205	118	207
16	2003	4	212	161	210	120	210
17	2003	5	213	198	207	174	212
18	2003	6	213	180	210	151	213
19	2003	7	209	175	209	120	206
20	2003	8	210	182	227	135	210
21	2003	9	212	81	209	60	210
22	2003	10	234	191	214	120	209
23	2003	11	210	181	210	60	210
24	2003	12	205	180	210	119	211
25	2004	1	210	198	207	123	209
26	2004	2	213	185	209	120	210
27	2004	3	209	173	147	117	209
28	2004	4	215	186	211	154	210
29	2004	5	209	179	210	139	210
30	2004	6	214	180	211	125	210
31	2004	7	210	64	131	139	208
32	2004	8	206	183	291	125	210
33	2004	9	207	177	206	124	208
34	2004	10	128	178	206	118	208
35	2004	11	215	176	213	120	208
36	2004	12	211	186	212	93	206
37	2005	1	208	151	205	98	215
38	2005	2	207	194	211	120	238
39	2005	3	206	180	212	124	208
40	2005	4	210	273	210	116	214
41	2005	5	211	281	208	123	208
42	2005	6	211	180	242	120	211
43	2005	7	260	183	213	122	206
44	2005	8	210	178	205	79	212
45	2005	9	211	180	208	120	136
46	2005	10	208	195	215	120	205
47	2005	11	207	178	211	88	207
48	2005	12	244	180	132	115	214
		Min	120	64	131	60	136
		Average	207	178	209	120	208
		Max	298	281	300	174	246
		Mod	210	180	210	120	210

F.4 Data collected for Time between Failures for Refineries

Refineries	TBF (minutes)				
	1	2	3	4	5
	Abadan	Arak	Bandar Abbas	Isfahab	Tehran
1	49800	28200	16200	17280	18000
2	27480	10680	39000	20040	14400
3	47400	6600	38820	54000	25200
4	9120	13200	18180	13860	27180
5	9180	17940	30660	51600	8820
6	29400	21900	58200	14700	31860
7	6600	10800	27000	23400	22800
8	27000	46200	8880	13800	50400
9	22800	12660	49800	8400	10800
10	10200	39600	8040	11400	11760
11	54000	10500	42600	58800	27480
12	16380	6480	50400	10740	11400
13	17820	9240	8580	9780	41400
14	8340	13020	22200	7500	19800
15	12780	40800	7920	9480	9540
16	22200	55800	17160	13560	43200
17	13680	6840	12840	9480	14520
18	7260	7920	45000	30000	16800
19	9240	14400	7200	23160	9720
20	47400	12000	49800	20700	11820
21	22440	28800	7560	39000	22500
22	14520	50400	42600	9600	33000

F.5 Data collected for Time to Repair for Refineries

Refineries	TTR (minutes)				
	1	2	3	4	5
	Abadan	Arak	Bandar Abbas	Isfahab	Tehran
1	60.0	180.0	30.0	60.0	168.0
2	90.0	138.0	60.0	102.0	120.0
3	138.0	60.0	162.0	90.0	120.0
4	192.0	60.0	312.0	108.0	240.0
5	66.0	324.0	72.0	324.0	264.0
6	84.0	240.0	126.0	108.0	60.0
7	66.0	216.0	30.0	162.0	228.0
8	198.0	342.0	78.0	168.0	30.0
9	126.0	318.0	60.0	192.0	108.0
10	210.0	84.0	63.0	126.0	21.0
11	30.0	114.0	108.0	192.0	24.0
12	270.0	216.0	168.0	354.0	36.0
13	594.0	306.0	138.0	324.0	132.0
14	138.0	432.0	72.0	246.0	66.0
15	210.0	132.0	264.0	312.0	318.0
16	36.0	186.0	96.0	450.0	144.0
17	228.0	228.0	108.0	312.0	192.0
18	120.0	264.0	9.0	96.0	57.0
19	42.0	360.0	180.0	114.0	240.0
20	384.0	150.0	156.0	390.0	264.0
21	276.0	330.0	216.0	180.0	84.0
22	288.0	198.0	120.0	384.0	132.0

F.6 Data collected for LPG Unloading time for Bps

	Unloading times (minutes)							
	Babol	Esfahan	Karaj	Kashan	Kerman	Mashahad	Qum	Yazd
1	159	87	96	132	94	222	141	117
2	123	74	111	156	81	988	137	146
3	123	82	98	101	122	983	121	118
4	174	156	126	140	123	113	92	133
5	76	105	143	161	134	737	131	131
6	139	107	120	149	108	954	138	127
7	119	166	137	158	118	627	96	107
8	79	142	104	125	133	130	121	95
9	94	132	135	123	97	565	91	129
10	100	76	117	165	120	194	146	102
11	143	110	131	146	90	1093	112	123
12	76	115	104	147	99	931	110	103
13	116	162	125	106	81	182	94	88
14	111	113	120	147	96	499	98	151
15	155	162	106	136	84	615	144	143
16	168	117	133	168	94	314	120	90
17	146	104	130	167	95	830	130	101
18	80	98	133	86	81	998	95	104
19	173	125	129	114	113	789	108	131
20	143	175	123	173	121	195	119	156
21	77	139	91	171	114	425	87	135
22	111	154	137	96	101	901	120	129
23	131	163	95	131	129	132	128	102
24	111	87	118	139	82	1071	104	89
25	113	124	98	121	97	442	129	124
26	141	94	97	99	93	380	87	114
27	78	109	116	92	96	706	106	95
28	84	88	135	130	82	252	86	157
29	163	92	122	97	106	912	115	142
30	96	96	126	159	113	930	116	155
31	105	78	123	89	129	344	141	139
32	124	112	99	174	121	615	145	108
33	169	154	101	132	117	1030	100	135
34	123	77	129	92	134	368	127	106
35	96	127	116	90	90	227	134	156
36	105	74	141	139	128	1015	143	129
37	160	104	113	153	99	445	148	135
38	145	173	132	122	103	430	122	103
39	115	174	108	112	115	235	101	96
40	86	75	99	152	88	470	138	96
41	172	127	107	113	113	91	118	96
42	116	128	121	115	116	426	100	150
43	109	68	105	143	129	879	146	118
44	155	164	122	151	120	1120	94	123
45	150	68	105	180	134	967	139	116
46	89	94	136	161	114	1025	92	96
47	86	105	138	96	103	357	146	118
48	140	80	133	180	133	101	132	138
49	96	134	103	136	84	700	128	148
50	166	98	102	170	84	637	126	87
Min	76	68	91	86	81	91	86	87
Averag	122	115	118	135	107	592	119	121
Max	174	175	143	180	134	1120	148	157

F.7 Data collected for bottling plants daily demands

The following data were only gathered for daily demand for Bottling plants during the month of March.

Appendix F: Data for Simulation Model Developments

Date	Karaj			Babol			Mashhad			Qom		
Mar-02	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit
1	0	0	41	0	0	0	0	0	0	0	0	0
2	1	113	77	11	20	27	55	19	4	12	20	27
3	71	55	60	0	20	34	70	0	11	5	20	34
4	49	56	0	122	20	16	59	0	8	0	20	16
5	0	0	70	0	0	0	0	0	0	0	0	0
6	45	56	77	128	39	17	50	0	11	0	39	17
7	31	75	49	115	19	17	39	0	10	17	19	17
8	28	56	47	140	20	19	29	0	11	19	20	19
9	36	154	84	0	0	0	0	0	0	0	0	0
10	144	56	64	161	18	21	18	18	11	20	18	21
11	116	95	76	137	20	20	25	0	10	17	20	20
12	147	132	0	138	58	22	15	19	10	16	58	22
13	0	0	57	0	0	0	0	0	0	0	0	0
14	204	55	67	120	19	17	23	19	13	53	19	17
15	202	19	52	136	20	20	29	19	12	55	20	20
16	154	77	90	0	0	0	0	0	0	0	0	0
17	180	55	82	117	19	19	36	0	12	55	19	19
18	144	38	74	139	19	20	24	0	12	55	19	20
19	100	57	65	103	0	17	12	19	9	55	0	17
20	83	37	58	105	19	16	22	19	13	38	19	16
21	54	74	67	87	0	13	28	0	10	41	0	13
22	70	76	32	54	37	14	18	18	12	28	37	14
23	79	73	52	0	0	0	0	0	0	0	0	0
24	120	119	76	128	0	20	24	0	11	52	0	20
25	180	95	77	125	38	18	128	18	11	30	38	18
26	198	56	68	129	19	17	20	0	11	51	19	17
27	178	56	61	129	0	14	9	18	9	53	0	14
28	166	56	75	142	19	15	19	18	10	39	19	15
29	160	18	25	147	19	14	27	76	11	44	19	14
30	136	114	90	0	0	0	0	0	0	0	0	0
31	192	37	0	135	20	16	93	38	10	49	20	16

Date	Karaj			Babol			Mashhad			Qom		
Mar-03	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit
1	0	134	27	0	39	23	0	0	11	0	0	0
2	0	196	52	71	0	0	104	0	13	0	39	22
3	107	96	46	87	0	18	99	0	16	0	57	27
4	75	75	65	87	20	33	93	0	16	15	19	21
5	125	77	60	70	56	35	85	0	16	46	19	22
6	136	75	70	57	39	40	74	19	5	44	19	17
7	153	77	0	78	0	0	63	0	15	40	0	0
8	158	38	80	0	38	37	0	19	17	0	19	29
9	235	57	83	78	20	35	71	18	19	43	19	20
10	192	73	80	76	20	29	80	0	19	33	19	21
11	167	36	77	58	71	0	86	18	5	32	19	20
12	160	56	0	43	0	40	73	0	19	30	0	0
13	119	75	80	0	54	0	0	0	5	0	57	22
14	176	38	31	86	0	28	76	0	20	29	19	0
15	171	75	83	0	20	42	0	0	16	0	19	31
16	179	92	93	100	0	36	62	19	16	63	19	29
17	171	38	80	90	55	26	48	0	18	52	19	21
18	170	93	61	49	20	26	55	0	15	42	19	18
19	129	74	74	68	0	23	44	18	15	40	19	19
20	160	111	70	61	96	0	31	18	5	41	0	18
21	161	196	40	36	0	34	39	0	17	40	19	0
22	202	132	100	0	38	35	0	0	17	0	19	19
23	182	57	72	108	56	29	47	18	18	42	19	18
24	215	95	60	112	0	32	35	0	15	41	19	18
25	200	113	70	133	20	33	42	38	15	42	19	17
26	235	75	78	104	56	34	29	0	15	43	19	15
27	279	56	76	92	20	0	57	56	5	45	0	19
28	276	76	35	115	0	40	47	0	17	49	19	0
29	257	93	75	0	0	40	0	0	19	0	19	22
30	297	38	120	100	55	0	93	19	5	49	0	30
31	316	0	0	61	0	0	82	0	0	46		0

Appendix F: Data for Simulation Model Developments

Date	Karaj			Babol			Mashhad			Qom		
Mar-04	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit
1	0	0	0	0	0	0	0	0	0	0	0	0
2	107	19	38	71	37	25	104	0	8	0	16	18
3	75	19	63	87	0	0	99	0	6	0	0	18
4	125	40	63	87	38	27	93	18	25	15	37	17
5	136	57	75	70	0	22	85	55	11	46	0	17
6	153	38	50	57	53	33	74	36	48	44	18	17
7	158	116	0	78	0	0	63	0	0	40	37	14
8	235	58	66	0	0	31	0	37	49	0	19	18
9	192	76	78	78	0	31	71	18	12	43	19	17
10	167	97	59	76	95	48	80	73	49	33	0	17
11	160	96	90	58	0	22	86	0	12	32	19	16
12	119	57	102	43	0	25	73	37	32	30	19	17
13	176	39	0	0	0	0	0	0	0	0	0	0
14	171	97	35	86	0	0	76	0	0	29	39	15
15	179	97	89	0	18	32	0	37	51	0	0	19
16	171	39	104	100	56	33	62	0	15	63	18	21
17	170	77	67	90	0	45	48	0	13	52	38	18
18	129	96	67	49	19	25	55	55	48	42	18	17
19	160	58	101	68	18	23	44	38	49	40	19	20
20	161	77	97	61	57	26	31	57	33	41	20	17
21	202	40	61	36	0	0	39	0	0	40	0	13
22	182	59	81	0	76	35	0	0	14	0	36	20
23	215	39	41	108	0	0	47	57	46	42	0	0
24	200	58	81	112	20	36	35	19	13	41	18	19
25	235	39	104	133	33	32	42	38	33	42	18	18
26	279	58	95	104	19	33	29	19	31	43	0	16
27	276	112	80	92	19	27	57	56	49	45	19	19
28	257	77	40	115	19	19	47	0	0	49	18	6
29	297	113	47	0	91	31	0	38	36	0	0	17
30	316	133	136	100	38	37	93	56	35	49	19	19
31		19	0	61	19	30	82	0	17	46	19	15

Date	Kashan			Yazd			Khayam			Esfahan			Kerman	
Mar-02	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance
1	0	0	0	0	0	0	0	0	0	67	40	0	0	0
2	19	18	20	29	20	24	0	19	0	106	59	71	0	0
3	17	37	22	24	22	48	0	0	0	95	39	69	19	56
4	32	18	27	0	28	0	19	0	0	65	96	84	50	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	24	18	20	48	20	38	18	0	0	78	119	86	44	19
7	22	18	22	38	21	28	0	18	0	110	58	85	46	0
8	19	18	19	28	19	36	18	38	0	84	77	88	33	74
9	0	0	0	0	0	0	18	0	0	72	58	53	0	0
10	18	18	19	36	18	27	36	0	0	77	78	100	91	18
11	18	37	24	27	23	17	0	0	0	55	78	91	91	36
12	31	18	25	17	25	26	0	18	0	41	95	93	109	0
13	0	0	0	0	22	0	19	0	0	0	0	0	0	0
14	25	0	22	26	11	17	0	19	0	43	117	85	98	36
15	3	57	11	17	0	24	18	38	0	75	79	97	118	17
16	0	0	0	0	26	0	18	0	0	57	39	58	0	0
17	48	0	26	24	26	31	0	19	0	39	79	94	123	19
18	22	38	26	31	25	21	35	0	0	24	114	96	126	0
19	34	0	25	21	18	30	17	0	0	42	75	82	107	19
20	9	37	19	29	20	20	18	0	0	36	58	53	104	18
21	28	18	20	20	18	10	0	0	0	40	97	66	107	16
22	59	18	18	10	0	18	18	38	0	71	79	80	109	0
23	0	0	0	0	19	0	18	0	0	70	39	40	0	0
24	26	19	19	18	18	8	0	19	0	68	76	83	91	0
25	26	18	18	7	19	17	18	19	0	62	78	70	68	18
26	26	19	19	17	18	7	19	37	0	68	58	59	71	18
27	26	18	18	74	17	16	18	0	0	67	19	87	73	18
28	26	17	17	16	17	8	0	0	0	34	57	62	7	18
29	27	17	17	8	0	19	18	0	0	57	57	76	80	19
30	0	0	0	0	18	0	0	0	0	9	58	35	0	0
31	47	18	18	19	0	9		18	0	31	77	66	88	0

Date	Kashan			Yazd			Khayam			Esfahan			Kerman	
Mar-03	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	27	19	20	94	63	40	0	0	0	52	96	60	149	18
3	26	38	20	60	59	66	0	0	0	88	39	86	27	16
4	45	0	22	72	72	87	0	0	0	41	39	39	38	0
5	22	37	22	63	63	75	19	18	18	41	112	91	33	0
6	38	19	21	56	55	63	0	0	0	62	77	67	17	18
7	36	19	21	85	85	79	18	18	18	73	57	61	22	18
8	0	0	0	0	0	0	0	0	0	69	77	75	0	0
9	34	0	23	95	95	89	18	18	18	70	115	100	29	17
10	10	37	22	104	104	83	18	18	18	85	78	87	31	54
11	28	19	23	96	96	95	36	36	36	76	113	104	72	0
12	22	38	22	87	86	94	0	0	0	85	95	74	40	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	38	37	24	95	95	83	19	18	18	106	135	132	32	0
15	0	0	0	0	0	0	0	0	0	110	19	37	0	0
16	51	0	23	87	86	8	18	18	18	92	78	86	16	18
17	28	19	23	96	96	9	18	18	18	83	114	102	21	18
18	23	55	21	87	86	8	0	0	0	95	76	75	23	55
19	57	19	20	78	78	8	35	35	35	96	77	67	45	0
20	55	0	18	88	88	7	17	17	17	105	75	81	26	0
21	37	38	20	99	99	0	18	18	18	99	91	67	13	92
22	0	0	0	0	0	7	0	0	0	122	56	57	0	0
23	54	0	20	92	92	7	18	18	18	121	77	64	92	18
24	34	19	20	84	84	7	18	18	18	134	57	78	93	0
25	33	19	19	95	95	7	0	0	0	114	57	70	59	0
26	33	19	19	88	87	7	18	18	18	100	56	70	50	0
27	33	19	18	98	98	7	19	18	18	87	58	79	42	0
28	34	0	18	91	91	0	18	18	18	65	76	73	28	0
29	0	19	0	0	0	7	0	0	0	68	57	33	0	0
30	34	15	18	102	102	12	18	18	18	92	94	122	19	0
31	35	0	25	95	94	0	0	0	0	65	38	59	5	37

Date	Kashan			Yazd			Khayam			Esfahan			Kerman	
Mar-04	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance	Exit	Inventory	Enterance
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	27	40	20	94	0	0	0	0	0	52	96	60	149	0
3	26	0	18	60	0	9	0	0	0	88	39	86	27	19
4	45	19	18	72	0	8	0	19	19	41	39	39	38	0
5	22	17	20	63	0	9	19	37	37	41	112	91	33	19
6	38	36	20	56	18	9	0	0	0	62	77	67	17	19
7	36	0	5	85	0	0	18	0	0	73	57	61	22	0
8	0	18	20	0	0	8	0	0	0	69	77	75	0	19
9	34	19	18	95	0	8	18	19	19	70	115	100	29	19
10	10	38	19	104	18	4	18	0	0	85	78	87	31	0
11	26	0	18	96	19	10	36	19	19	76	113	104	72	0
12	22	19	19	87	0	9	0	36	36	85	95	74	40	19
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	38	19	12	95	0	0	19	0	0	106	135	132	32	0
15	0	18	20	0	0	9	0	0	0	110	19	37	0	19
16	51	18	20	87	0	9	18	18	18	92	78	86	16	0
17	28	38	20	96	0	8	18	19	19	83	114	102	21	19
18	23	0	20	87	19	4	0	0	0	95	76	75	23	19
19	57	17	20	78	0	9	35	18	18	96	77	67	45	19
20	55	20	20	88	19	9	17	36	36	105	75	81	26	19
21	37	0	8	99	0	0	18	0	0	99	91	67	13	0
22	0	36	20	0	0	9	0	37	37	122	56	57	0	19
23	54	0	0	92	0	0	18	0	0	121	77	64	92	0
24	34	37	20	84	19	10	18	19	19	134	57	78	93	19
25	33	0	18	95	0	8	0	0	0	114	57	70	59	38
26	33	0	20	88	0	9	18	0	0	100	56	70	50	19
27	33	34	20	98	0	8	19	38	38	87	58	79	42	19
28	34	17	9	91	0	0	18	0	0	65	76	73	28	0
29	0	37	20	0	18	8	0	37	37	68	57	33	0	20
30	34	0	21	102	19	8	18	0	0	92	94	122	19	19
31	35	18	20	95	0	6	0	0	0	65	38	59	5	19

F.8 Data collected for customers' arrival pattern in Bottling plants

Babol		
Shift Periods	Records	Model %
390+270	74	25.05
	196	
	80	
660+30	173	12.38
690+240	170	17.90
	80	
930+90	42	37.51
	90	
	196	
	196	
1020+120	100	7.16

Kashan		
Shift Periods	Records	Model %
390+270	100	32.05
	200	
	200	
	40	
660+30	200	11.87
690+240	105	6.23
930+90	230	28.49
	47	
	102	
	101	
1020+120	100	21.36
	90	
	38	
	62	
	70	

Mashhad		
Shift Periods	Records	Model %
390+270	70	31.67
	102	
	73	
	100	
	110	
	32	
	106	
	87	
660+30	70	11.18
	170	
690+240	110	9.32
	90	
930+90	106	43.60
	72	
	688	
	70	
1020+120	21	4.24
	70	

Qom		
Shift Periods	Records	Model %
390+270	260	30.18
	174	
660+30	180	12.52
690+240	80	17.11
	166	
930+90	110	28.51
	300	
1020+120	88	11.68
	80	

Yazd		
Shift Periods	Records	Model %
390+270	20	30.33
	155	
660+30	70	12.13
690+240	140	24.26
930+90	122	21.14
1020+120	70	12.13

F.9 Historical Data on LPG Filling Machines in Bottling Plants

The following summarises data collected to establish distributions for Mean Time To Failure and Mean Time To Repair for LPG filling machines across the bottling plants.

F.9.1 Babol Bottling Plant:

Babol	Filling Machines	TBF (minutes)	T R (minutes)
1	S1	9096	44
2	S2	9684	52
3	S3	11928	49
4	S4	11964	22
5	S5	8580	51
6	S6	11280	19
7	S7	9372	36
8	S8	6444	39
9	S9	9468	13
10	S10	7344	36
11	S11	7704	20
12	S12	5604	23

F.9.2 Esfahan Bottling Plants:

Esfahan	Filling Machines	TBF (minutes)	T R (minutes)
1	S1	8544	28
2	S2	6528	34
3	S3	10212	38
4	S4	8556	27
5	S5	10800	32
6	S6	8448	43
7	S7	8796	33
8	S8	10740	51
9	S9	11256	29
10	S10	9204	37
11	S11	8028	47
12	S12	8652	19
13	S13	9900	25
14	S14	10140	15
15	S15	6672	55
16	S16	8124	67
17	S17	11976	25
18	S18	10572	19
19	S19	7236	32
20	S20	9552	24
21	S21	8508	47
22	S22	9792	23
23	S23	6492	15
24	S24	11052	29

F.9.3 Karaj Bottling Plants:

Karaj	Filling Machines	TBF (minutes)	T R (minutes)
1	S1	5028	38
2	S2	10668	33
3	S3	8880	51
4	S4	9660	42
5	S5	6288	19
6	S6	5736	26
7	S7	11472	60
8	S8	10200	21
9	S9	10068	42
10	S10	8604	23
11	S11	6396	10
12	S12	5664	19
13	S13	7956	13
14	S14	6840	23
15	S15	6408	23
16	S16	9024	36
17	S17	5952	32
18	S18	11292	28
19	S19	9504	26
20	S20	6540	17
21	S21	6048	45
22	S22	9828	41
23	S23	10632	28
24	S24	11688	24

F.9.4 Kashan Bottling Plants:

Kashan	Filling Machines	TBF (minutes)	T R (minutes)
1	S1	8700	16
2	S2	8928	37
3	S3	8028	35
4	S4	8292	16
5	S5	10656	35
6	S6	8304	26
7	S7	11088	24
8	S8	10992	16

F.9.5 Kerman Bottling Plants:

kerman	Filling Machines	TBF (minutes)	T R (minutes)
1	S1	8088	24
2	S2	7452	21
3	S3	9504	31
4	S4	10980	15
5	S5	7380	15
6	S6	11400	18
7	S7	5964	14
8	S8	11844	17

F.9.6 Mashhad Bottling Plants:

Mashhad	Filling Machines	TBF (minutes)	T R (minutes)
1	S1	9732	11
2	S2	7044	12
3	S3	7836	25
4	S4	10116	18
5	S5	9444	25
6	S6	11868	39
7	S7	6300	32
8	S8	8976	35

F.9.7 Qom Bottling Plants:

Qom	Filling Machines	TBF (minutes)	T R (minutes)
1	S1	9900	33
2	S2	11292	19
3	S3	5544	21
4	S4	8892	31
5	S5	8424	20
6	S6	8568	14
7	S7	11328	19
8	S8	10272	15

F.9.8 Yazd Bottling Plants:

Yazd	Filling Machines	TBF (minutes)	T R (minutes)
1	S1	7356	22
2	S2	11568	29
3	S3	5736	19
4	S4	11172	10

F.10 Data collected for Cylinder Inspection Area

Babol			
Data	Record	Rejects	Accept
1	3454	173	3281
2	1818	91	1727
3	3727	187	3540
4	2545	128	2417
5	2181	110	2071
6	3363	169	3194
7	3272	164	3108
8	3090	155	2935
9	2636	132	2504
10	3090	155	2935
11	2454	123	2331
12	2090	105	1985
13	3000	150	2850
14	2454	123	2331
15	3181	160	3021
16	2727	137	2590
17	3000	150	2850
18	2909	146	2763
19	2909	146	2763
20	2090	105	1985
21	2181	110	2071
22	3909	196	3713
23	2727	137	2590
24	3545	178	3367
25	3272	164	3108
26	2818	141	2677
27	3272	164	3108
28	1818	91	1727
29	4090	205	3885
30	2727	137	2590
Total	86349	4332	82017
Average %		5.02	94.98

Esfahan			
Data	Record	Rejects	Accept
1	10272	462	9810
2	3454	155	3299
3	3454	155	3299
4	5090	229	4861
5	5272	237	5035
6	6272	282	5990
7	3545	159	3386
8	5363	241	5122
9	6909	310	6599
10	8818	396	8422
11	5272	237	5035
12	8727	392	8335
13	10363	466	9897
14	7000	315	6685
15	7000	315	6685
16	3454	155	3299
17	5272	237	5035
18	5272	237	5035
19	7000	315	6685
20	6909	310	6599
21	7000	315	6685
22	10545	474	10071
23	8818	396	8422
24	6909	310	6599
25	5454	245	5209
26	8818	396	8422
27	10454	470	9984
28	4636	208	4428
29	6363	286	6077
30	3363	151	3212
Total	197078	8868.51	188222
Average %		4.49	95.51

Karaj			
Data	Record	Rejects	Accept
1	7272	360	6912
2	4636	231	4405
3	6363	317	6046
4	3363	168	3195
5	5272	260	5012
6	6272	313	5959
7	5363	265	5098
8	5545	278	5267
9	8818	440	8378
10	7545	375	7170
11	7454	370	7084
12	7545	376	7169
13	7636	381	7255
14	6545	320	6225
15	9545	400	9145
16	7272	363	6909
17	10909	545	10364
18	5454	272	5182
19	6272	313	5959
20	4272	213	4059
21	8636	431	8205
22	7545	377	7168
23	6909	345	6564
24	10000	483	9517
25	6545	320	6225
26	8545	400	8145
27	7818	386	7432
28	5454	264	5190
29	7818	378	7440
30	7000	324	6676
Total	209623	10268	199355
Average %		4.90	95.10

Kashan			
Data	Record	Rejects	Accept
1	1909	95	1814
2	1181	59	1122
3	1909	95	1814
4	909	45	864
5	1454	72	1382
6	1727	86	1641
7	1636	81	1555
8	1636	81	1555
9	1727	86	1641
10	1818	90	1728
11	1818	90	1728
12	818	40	778
13	2000	100	1900
14	1454	72	1382
15	1818	90	1728
16	1727	86	1641
17	636	31	605
18	1727	86	1641
19	1818	90	1728
20	1727	86	1641
21	818	40	778
22	727	36	691
23	1909	95	1814
24	1363	68	1295
25	1818	90	1728
26	1090	54	1036
27	1000	50	950
28	727	36	691
29	1727	86	1641
30	1727	86	1641
Total	44355	2217.75	42153
Average %		4.96	95.04

Kerman			
Data	Record	Rejects	Accept
1	1727	86	1641
2	2181	109	2072
3	2181	109	2072
4	909	45	864
5	1454	72	1382
6	2000	100	1900
7	1454	72	1382
8	1727	86	1641
9	1545	77	1468
10	1727	86	1641
11	1545	77	1468
12	2181	109	2072
13	1000	50	950
14	1363	68	1295
15	1272	63	1209
16	909	45	864
17	1000	50	950
18	1181	59	1122
19	1000	50	950
20	1545	77	1468
21	1636	81	1555
22	1181	59	1122
23	1000	50	950
24	1454	72	1382
25	1090	54	1036
26	1363	68	1295
27	1727	86	1641
28	2181	109	2072
29	1454	72	1382
30	1454	72	1382
Total	44441	2222.05	42228
Average %		4.98	95.02

Mashhad			
Data	Record	Rejects	Accept
1	2727	135	2592
2	1818	90	1728
3	3727	185	3542
4	1818	90	1728
5	5000	249	4751
6	3181	158	3023
7	1636	81	1555
8	5000	249	4751
9	1727	86	1641
10	3181	158	3023
11	5272	262	5010
12	3272	162	3110
13	3000	149	2851
14	5000	249	4751
15	5272	262	5010
16	1636	81	1555
17	1636	81	1555
18	5090	253	4837
19	3454	172	3282
20	5090	253	4837
21	1818	90	1728
22	5090	253	4837
23	5272	262	5010
24	3545	176	3369
25	3272	162	3110
26	1636	81	1555
27	2636	131	2505
28	2545	126	2419
29	3454	172	3282
30	2000	99	1901
Total	99805	4957	94848
Average %		4.97	95.03

Qom			
Data	Record	Rejects	Accept
1	2363	118	2245
2	1363	68	1295
3	1363	68	1295
4	545	27	518
5	1363	68	1295
6	1454	72	1382
7	1363	68	1295
8	1272	63	1209
9	1545	77	1468
10	1727	86	1641
11	1636	81	1555
12	1636	81	1555
13	1545	77	1468
14	1727	86	1641
15	1545	77	1468
16	1818	90	1728
17	636	31	605
18	1000	50	950
19	1909	95	1814
20	1545	77	1468
21	1454	72	1382
22	1545	77	1468
23	1636	81	1555
24	545	27	518
25	1454	72	1382
26	1272	63	1209
27	1545	77	1468
28	1363	68	1295
29	1181	59	1122
30	1363	68	1295
Total	42713	2124	40589
Average %		4.97	95.03

Yazd			
Data	Record	Rejects	Accept
1	818	40	778
2	545	27	518
3	636	31	605
4	636	31	605
5	636	31	605
6	545	27	518
7	636	31	605
8	636	31	605
9	545	27	518
10	545	27	518
11	636	31	605
12	545	27	518
13	636	31	605
14	636	31	605
15	636	31	605
16	545	27	518
17	636	31	605
18	545	27	518
19	636	31	605
20	636	31	605
21	636	31	605
22	636	31	605
23	727	36	691
24	636	31	605
25	636	31	605
26	636	31	605
27	636	31	605
28	727	36	691
29	636	31	605
30	727	36	691
Total	18898	944.9	17972
Average %		4.90	95.10

F.11 Data collected for the cylinder maintenance duration in different Bottling plants

Record	Rework time per cylinder (Min)							
	Karaj	Babol	Mashhad	Qom	Kashan	Yazd	Kerman	Esfahan
1	3.60	4.94	3.00	3.93	3.17	4.00	2.15	5.78
2	1.68	2.60	2.00	2.27	3.93	2.70	2.73	1.94
3	2.31	5.34	2.00	2.27	3.17	3.10	2.73	1.94
4	3.17	3.66	4.11	2.70	4.50	3.10	2.25	2.86
5	2.60	3.14	5.53	2.27	2.40	3.10	3.60	2.96
6	3.13	4.83	3.51	2.40	2.87	2.70	5.00	3.53
7	2.65	4.69	1.80	2.27	2.70	3.10	3.60	1.99
8	2.78	4.43	5.53	2.10	2.70	2.70	4.30	3.01
9	4.40	3.77	1.91	2.57	2.87	2.70	3.85	3.88
10	3.81	4.43	3.51	2.87	3.00	3.10	4.30	4.95
11	3.76	3.51	3.60	2.70	3.00	3.10	3.85	2.96
12	3.75	3.00	5.82	2.70	2.67	2.70	5.45	4.90
13	3.70	4.29	3.31	2.57	3.33	3.10	2.50	5.83
14	3.20	4.57	5.53	2.87	2.40	3.10	3.40	3.94
15	4.00	4.29	5.82	2.57	3.00	3.10	3.15	3.94
16	3.63	3.91	1.80	3.00	2.87	2.70	2.25	1.94
17	5.45	3.51	1.80	2.07	5.73	3.10	2.50	2.96
18	2.72	4.17	5.62	3.33	3.00	2.70	2.95	2.96
19	3.13	4.17	3.82	3.17	2.87	3.10	2.50	3.94
20	2.13	3.00	5.62	2.57	2.67	3.10	3.85	3.88
21	4.31	3.14	2.00	2.40	2.07	3.10	4.05	3.94
22	3.77	5.60	5.62	2.57	2.40	3.10	2.95	5.93
23	3.45	3.91	5.82	2.70	3.17	3.60	2.50	4.95
24	4.83	5.09	3.91	2.70	6.80	3.10	3.60	3.88
25	3.20	4.69	3.60	2.40	3.00	3.10	2.70	1.89
26	4.00	5.86	1.80	2.10	3.60	3.10	3.40	2.60
27	3.86	4.69	2.91	2.57	3.33	3.10	4.30	3.06
28	2.64	4.03	2.80	2.27	2.40	3.60	5.45	3.58
29	3.78	2.60	3.82	2.95	4.30	3.10	3.60	4.95
30	3.24	3.91	2.20	2.27	2.87	3.60	3.60	5.88
Average	3.42	4.13	3.67	2.60	3.23	3.09	3.44	3.69

F.12 Speed Parameters For Unloaded Trucks

Route	From	To	Average Speed (time taken to reach the destination / Distance)										Result		
			Record 1	Record 2	Record 3	Record 4	Record 5	Record 6	Record 7	Record 8	Record 9	Record 10	Min	Average	Max
1	Abadan	Babol	109	96	98	80	94	91	90	92	98	93	80	94.11	109
2	Abadan	Isfahan	98	103	99	95	98	96	100	98	92	98	92	97.71	103
3	Abadan	Karaj	90	103	98	91	90	113	91	118	98	103	90	99.51	118
4	Abadan	Kashan	96	96	93	104	98	101	103	96	115	98	93	100.04	115
5	Abadan	Kerman	98	95	92	103	93	106	96	98	97	101	92	97.90	106
6	Abadan	Mashhad	107	112	92	96	88	91	91	95	98	103	88	97.31	112
7	Abadan	Qom	112	106	114	108	108	103	109	98	108	101	98	106.65	114
8	Abadan	Yazd	96	95	108	106	103	103	106	92	104	113	92	102.60	113
9	Bandar Abbas	Babol	103	90	88	93	117	98	110	96	112	100	88	100.60	117
10	Bandar Abbas	Isfahan	110	90	109	95	95	104	104	102	98	96	90	100.39	110
11	Bandar Abbas	Karaj	92	93	102	114	102	90	99	114	98	105	90	100.86	114
12	Bandar Abbas	Kashan	111	82	92	103	96	96	85	98	105	111	82	97.90	111
13	Bandar Abbas	Kerman	96	110	90	93	96	93	93	88	91	80	80	92.92	110
14	Bandar Abbas	Mashhad	93	103	87	100	103	88	98	107	98	96	87	97.24	107
15	Bandar Abbas	Qom	81	104	98	103	91	92	104	98	101	98	81	96.98	104
16	Bandar Abbas	Yazd	92	90	104	98	111	100	84	103	96	108	84	98.58	111
17	Arak	Babol	84	88	88	104	84	83	86	109	86	83	83	89.46	109
18	Arak	Isfahan	92	77	98	86	88	94	87	96	100	96	77	91.39	100
19	Arak	Karaj	95	83	90	94	88	92	110	90	85	80	80	90.71	110
20	Arak	Kashan	86	96	72	100	87	88	86	88	81	90	72	87.43	100
21	Arak	Kerman	93	88	98	89	102	96	92	88	103	84	84	93.34	103
22	Arak	Mashhad	110	86	110	101	103	99	95	98	96	98	86	99.59	110
23	Arak	Qom	92	99	90	105	90	85	75	89	83	89	75	89.71	105
24	Arak	Yazd	100	98	95	95	93	87	92	88	88	88	87	92.38	100
25	Isfahan	Babol	108	84	76	108	90	92	98	90	82	99	76	92.62	108
26	Isfahan	Isfahan	90	84	96	86	90	92	105	90	88	85	84	90.64	105
27	Isfahan	Karaj	98	98	94	105	100	106	100	92	98	98	92	98.88	106
28	Isfahan	Kashan	85	77	88	84	88	93	90	105	84	86	77	87.95	105
29	Isfahan	Kerman	95	86	94	88	90	89	84	110	85	87	84	90.83	110
30	Isfahan	Mashhad	105	100	98	94	95	94	92	104	95	96	92	97.31	105
31	Isfahan	Qom	108	99	100	89	89	93	92	93	88	90	88	94.13	108
32	Isfahan	Yazd	80	111	83	86	93	96	90	88	85	83	80	89.42	111
33	Tehran	Babol	83	86	95	100	76	86	87	80	78	74	74	84.56	100
34	Tehran	Isfahan	96	104	90	98	93	106	110	108	103	102	90	101.08	110
35	Tehran	Karaj	96	92	88	84	90	76	86	102	94	89	76	89.74	102
36	Tehran	Kashan	87	103	86	88	84	70	96	88	90	88	70	87.96	103
37	Tehran	Kerman	98	79	93	87	116	106	90	88	96	89	79	94.10	116
38	Tehran	Mashhad	94	92	105	98	103	109	96	94	98	113	92	100.22	113
39	Tehran	Qom	84	82	87	102	88	81	88	84	86	93	81	87.50	102
40	Tehran	Yazd	95	95	93	87	96	100	87	88	91	88	87	92.02	100

F.13 Speed Parameters for Loaded trucks

Route	From	To	Average Speed (time taken to reach the destination / Distance)										Result		
			Record 1	Record 2	Record 3	Record 4	Record 5	Record 6	Record 7	Record 8	Record 9	Record 10	Min	Average	Max
1	Abadan	Babol	80	78	74	70	76	73	72	74	78	75	70	75.00	80
2	Abadan	Isfahan	80	85	81	77	80	78	82	80	74	82	74	79.90	85
3	Abadan	Karaj	74	85	80	82	72	78	80	84	80	85	72	80.00	85
4	Abadan	Kashan	78	80	75	86	80	83	85	74	81	80	74	80.20	86
5	Abadan	Kerman	80	77	74	85	75	88	78	80	79	83	74	79.90	88
6	Abadan	Mashhad	89	84	74	78	72	80	85	77	80	81	72	80.00	89
7	Abadan	Qom	76	88	85	90	92	85	82	80	90	83	76	85.10	92
8	Abadan	Yazd	78	80	90	88	85	85	88	74	90	95	74	85.30	95
9	Bandar Abbas	Babol	85	72	76	75	81	80	85	78	80	85	72	79.70	85
10	Bandar Abbas	Isfahan	85	80	80	77	77	82	74	85	80	80	74	80.00	85
11	Bandar Abbas	Karaj	74	75	84	77	80	72	81	88	80	87	72	79.80	88
12	Bandar Abbas	Kashan	81	75	74	85	78	78	80	80	87	86	74	80.40	87
13	Bandar Abbas	Kerman	78	82	72	75	78	80	75	70	73	70	70	75.30	82
14	Bandar Abbas	Mashhad	75	85	79	82	85	70	80	88	80	78	70	80.20	88
15	Bandar Abbas	Qom	73	86	80	85	80	74	80	82	83	80	73	80.30	86
16	Bandar Abbas	Yazd	74	71	90	80	80	82	73	85	78	89	71	80.20	90
17	Arak	Babol	65	70	70	74	66	65	68	75	72	76	65	70.10	76
18	Arak	Isfahan	74	72	80	74	70	76	73	72	82	78	70	75.10	82
19	Arak	Karaj	77	65	72	76	70	70	65	72	67	70	65	70.40	77
20	Arak	Kashan	68	72	66	72	69	70	68	70	69	72	66	69.60	72
21	Arak	Kerman	75	70	80	77	84	78	74	70	76	70	70	75.40	84
22	Arak	Mashhad	78	77	80	84	85	81	77	80	78	80	77	80.00	85
23	Arak	Qom	74	70	72	73	72	67	68	71	65	68	65	70.00	74
24	Arak	Yazd	82	80	77	77	75	69	74	70	72	70	69	74.60	82
25	Isfahan	Babol	74	67	65	76	72	74	70	72	64	70	64	70.40	76
26	Isfahan	Isfahan	72	66	72	68	72	74	77	65	70	67	65	70.30	77
27	Isfahan	Karaj	80	82	76	87	70	88	82	79	80	80	70	80.40	88
28	Isfahan	Kashan	67	70	70	73	70	75	72	70	66	68	66	70.10	75
29	Isfahan	Kerman	77	68	69	70	72	71	66	67	70	69	66	69.90	77
30	Isfahan	Mashhad	87	82	80	83	77	76	74	80	77	78	74	79.40	87
31	Isfahan	Qom	80	74	82	71	72	75	74	75	70	72	70	74.50	82
32	Isfahan	Yazd	70	77	65	68	75	78	72	70	67	65	65	70.70	78
33	Tehran	Babol	65	68	70	62	60	68	69	62	60	63	60	64.70	70
34	Tehran	Isfahan	78	75	79	80	77	85	80	82	85	84	75	80.50	85
35	Tehran	Karaj	78	68	70	66	72	64	68	72	70	71	64	69.90	78
36	Tehran	Kashan	69	74	68	70	66	64	69	70	72	70	64	69.20	74
37	Tehran	Kerman	80	72	75	69	80	78	70	70	78	79	69	75.10	80
38	Tehran	Mashhad	76	74	88	80	85	82	78	77	80	82	74	80.20	88
39	Tehran	Qom	66	70	69	74	70	67	70	66	68	75	66	69.50	75
40	Tehran	Yazd	77	77	75	69	78	82	72	70	73	70	69	74.30	82

F.14 Trucks Activities Pattern

The following tables illustrate trucks working pattern. This working pattern is collected for a three months period. The working pattern is used to establish the time between failures and time to repair for the trucks in the simulation model.

Status	Descriptions
O	Trucks off works but available
R	Trucks taken for repairs

	W1							W2							W3						
Truck	D1	D2	D3	D4	D5	D6	D7	D1	D2	D3	D4	D5	D6	D7	D1	D2	D3	D4	D5	D6	D7
T1	366	368	326	O	O	O	O	502	497	310	282	321	349	O	437	336	387	466	322	380	454
T2	453	453	368	326	277	281	407	O	O	414	381	312	431	O	396	427	490	415	290	305	O
T3	477	450	R	R	R	R	O	402	215	388	462	247	484	O	402	470	496	293	370	359	O
T4	373	298	213	321	248	415	O	O	O	O	O	O	O	O	468	400	311	444	302	341	O
T5	435	430	392	349	502	330	O	378	409	346	437	412	299	O	343	389	342	O	O	O	362
T6	O	O	O	506	489	274	O	300	442	448	322	466	394	O	460	430	R	R	R	369	O
T7	468	459	335	437	380	264	O	342	473	278	442	265	329	O	290	334	378	303	431	299	O
T8	288	499	463	336	490	R	R	O	316	233	429	390	396	O	370	379	300	473	484	394	O
T9	480	349	366	449	272	399	O	366	246	329	388	361	402	O	302	430	342	308	369	329	O
T10	312	411	347	537	464	435	O	450	237	412	431	458	468	R	R	R	522	431	299	376	O
T11	396	448	312	272	361	474	O	O	O	O	O	368	378	O	414	396	437	429	289	299	O
T12	497	327	387	464	355	453	O	240	295	347	448	473	300	O	396	240	322	388	337	449	R
T13	357	437	366	361	O	O	O	234	292	522	367	464	342	O	240	456	450	431	346	297	O
T14	399	336	453	355	500	298	O	414	245	328	364	O	O	522	456	420	369	380	382	301	O
T15	O	O	O	O	O	O	O	396	427	336	350	361	366	O	420	490	343	460	388	398	O
T16	323	537	373	500	472	457	O	240	470	377	482	246	450	O	390	496	389	430	354	395	O
T17	268	272	435	349	401	459	O	456	400	490	424	427	354	O	O	O	O	O	O	O	O
T18	324	464	328	411	382	499	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
T19	436	361	468	448	410	349	O	390	473	311	413	436	234	O	368	484	395	305	395	306	O
T20	282	298	485	327	382	497	O	361	308	431	340	215	414	O	473	369	342	343	315	342	O
T21	448	388	368	280	347	310	R	458	431	484	425	319	396	O	295	299	324	460	449	324	O
T22	348	R	R	435	338	282	509	O	394	369	362	409	240	O	367	394	366	362	379	366	O
T23	368	448	382	401	373	321	R	473	472	299	378	259	456	O	414	329	419	448	440	419	O
T24	326	348	373	417	266	349	O	464	343	394	349	255	420	O	388	376	303	420	337	398	345
T25	277	469	387	327	379	506	O	393	381	329	423	380	366	487	412	299	400	O	O	O	O
T26	281	O	O	O	O	O	O	361	462	376	307	322	450	O	346	349	444	296	411	392	O
T27	407	299	322	404	327	336	O	246	347	299	298	317	354	O	448	442	334	R	R	R	R
T28	241	348	380	450	365	449	O	427	437	349	434	289	240	O	278	429	379	412	303	O	O
T29	502	256	454	O	O	O	O	O	O	442	224	352	234	O	233	388	430	349	400	O	O
T30	497	240	323	309	277	272	O	424	450	429	259	295	414	444	329	431	R	O	O	371	O
T31	310	293	268	351	436	464	O	418	369	388	255	367	396	O	427	380	440	292	245	310	O
T32	282	390	497	420	O	O	O	413	259	431	380	215	240	O	400	448	337	336	377	420	O
T33	321	323	357	417	298	355	O	340	255	380	322	319	456	O	436	367	487	466	265	325	O
T34	349	268	399	327	347	418	O	278	380	448	R	R	347	O	215	364	321	435	400	296	O
T35	506	324	435	373	361	500	O	233	322	367	412	429	522	O	319	350	394	337	444	411	O
T36	437	O	O	O	O	217	O	329	317	364	346	388	328	O	378	482	415	392	463	448	O
T37	336	282	453	450	418	472	O	412	289	350	448	431	336	O	349	463	420	361	376	336	O
T38	449	448	450	418	500	401	O	387	352	482	278	442	377	O	423	376	325	392	361	347	O
T39	O	O	O	O	217	382	388	O	O	396	233	429	415	O	307	394	296	325	296	304	O
T40	258	505	430	217	600	410	456	O	O	349	423	307	298	O	298	444	411	306	293	263	O

	W4							W5							W6						
Truck	D1	D2	D3	D4	D5	D6	D7	D1	D2	D3	D4	D5	D6	D7	D1	D2	D3	D4	D5	D6	D7
T1	O	O	O	502	497	310	O	380	454	R	R	R	R	R	474	453	450	298	430	457	O
T2	323	268	324	R	R	R	O	440	413	362	324	290	289	O	277	281	407	354	419	387	380
T3	413	324	384	404	353	444	O	337	324	O	O	O	337	O	468	319	412	347	442	369	O
T4	362	307	280	485	331	398	O	487	384	280	404	343	346	O	307	384	305	412	400	337	R
T5	307	280	485	331	398	382	404	321	404	485	353	460	382	O	449	379	440	349	444	346	O
T6	395	289	289	413	362	444	O	394	353	331	444	362	388	O	O	O	O	371	463	382	404
T7	342	337	R	R	R	302	O	415	444	398	348	448	354	O	O	472	324	310	376	388	353
T8	324	346	306	460	396	341	O	290	348	R	R	379	388	O	O	449	366	420	394	354	444
T9	O	O	O	O	412	342	O	305	471	R	290	424	368	O	366	382	293	325	444	388	O
T10	419	388	370	448	349	R	O	343	472	395	305	334	R	R	310	343	293	370	302	368	O
T11	303	400	359	379	371	430	O	460	449	370	343	379	325	O	420	460	444	302	341	395	O
T12	R	R	389	424	310	428	O	362	297	359	460	430	296	O	325	362	395	444	342	449	O
T13	400	289	301	334	420	335	O	448	301	389	O	O	O	O	O	362	374	411	396	297	O
T14	359	337	398	379	325	278	O	379	449	430	448	455	392	O	321	318	376	O	306	301	O
T15	389	346	395	430	O	O	376	424	379	428	379	367	430	O	454	449	477	418	217	430	O
T16	430	382	418	O	411	337	O	334	440	335	424	403	421	O	361	311	342	396	388	418	O
T17	428	388	518	370	302	392	O	379	337	278	334	433	394	O	458	431	398	290	368	318	O
T18	335	R	R	R	R	361	O	430	487	435	379	440	413	O	420	303	496	418	400	240	O
T19	278	368	297	342	392	424	O	335	321	O	430	319	368	O	296	460	362	448	379	424	O
T20	293	395	301	396	382	428	O	278	O	O	412	407	323	375	411	430	428	335	278	435	O
T21	370	315	398	412	388	400	O	435	O	O	349	305	317	415	392	396	412	349	371	310	O
T22	359	449	395	349	354	444	O	337	R	R	371	375	396	357	430	370	R	R	R	506	471
T23	389	379	418	371	388	463	O	392	444	321	310	311	408	O	421	430	374	455	367	403	O
T24	O	O	318	310	368	376	O	361	302	394	420	322	290	O	394	361	424	428	448	336	O
T25	460	362	306	379	424	394	379	424	341	415	379	413	448	O	413	296	411	392	430	421	O
T26	430	428	335	278	435	444	O	428	342	290	440	324	379	O	447	466	373	304	437	289	O
T27	396	412	349	371	310	420	325	343	396	305	337	384	424	O	537	348	298	500	356	337	O
T28	310	424	435	305	256	317	O	460	412	343	487	404	334	O	436	474	404	355	440	346	O
T29	420	334	337	375	281	396	O	362	349	460	321	353	379	O	263	319	263	368	500	382	O
T30	325	379	392	311	290	O	O	O	O	362	394	444	430	550	382	407	382	323	365	388	O
T31	296	430	361	322	341	290	O	424	395	444	415	348	374	O	256	305	256	317	R	R	R
T32	411	374	424	334	337	420	O	334	418	463	290	471	455	O	281	336	347	304	263	382	O
T33	392	455	428	379	392	325	O	379	318	376	305	472	367	O	290	421	394	413	368	323	O
T34	430	O	O	O	O	O	O	O	O	O	O	O	403	O	394	448	430	361	296	430	306
T35	421	403	336	374	424	411	O	374	293	444	411	297	433	O	413	O	O	388	353	331	O
T36	394	433	347	455	428	O	O	455	R	302	392	301	440	O	368	514	R	354	444	398	O
T37	413	440	304	367	448	430	O	367	359	341	430	430	319	O	323	289	413	362	317	O	O
T38	O	319	263	403	336	421	O	403	389	342	421	361	407	O	317	337	324	307	396	R	R
T39	440	319	407	305	375	392	O	433	430	396	394	296	305	O	396	346	384	280	408	O	O
T40	304	263	382	256	281	408	O	368	421	394	413	R	R	O	408	382	404	485	290	374	O

W7							W8							W9						
D1	D2	D3	D4	D5	D6	D7	D1	D2	D3	D4	D5	D6	D7	D1	D2	D3	D4	D5	D6	D7
499	349	411	448	327	280	O	401	417	327	373	404	450	O	309	351	457	459	499	R	O
328	457	229	270	280	502	337	310	303	413	502	402	427	O	321	394	302	415	444	366	O
O	O	O	390	404	497	388	400	431	362	497	215	490	R	R	R	R	379	430	450	O
R	R	R	R	R	310	388	348	369	444	310	414	415	O	450	366	424	300	342	240	O
367	440	315	402	310	282	O	O	O	394	282	381	290	O	237	246	388	431	429	379	O
403	319	449	215	380	321	O	390	395	353	321	312	305	O	329	233	468	299	289	300	O
433	407	379	414	454	349	O	415	289	343	349	431	413	O	388	429	343	329	376	431	O
490	305	424	381	315	437	O	315	289	460	437	396	324	O	361	390	430	419	303	299	O
415	375	334	312	449	336	O	479	O	O	O	O	378	O	402	396	240	R	R	329	O
290	311	379	431	379	387	O	400	331	379	404	382	409	O	370	290	295	341	342	419	O
305	322	430	396	378	466	O	311	398	354	440	343	388	O	394	334	412	290	305	388	289
413	413	374	427	409	322	O	444	382	336	413	460	462	O	302	378	431	348	471	370	376
324	362	455	337	388	380	O	302	321	387	362	362	247	O	415	308	458	395	370	460	O
384	324	353	324	462	323	O	O	O	O	324	362	484	478	444	369	468	305	343	396	O
440	290	444	485	247	268	O	341	404	466	290	318	402	O	415	394	341	376	342	302	O
R	343	R	R	484	324	O	307	O	O	280	449	470	O	290	324	290	303	305	430	O
293	337	392	325	402	502	O	280	398	322	O	O	O	O	O	346	348	400	471	342	343
370	278	361	296	470	497	O	485	348	380	343	404	293	O	343	306	395	359	370	429	O
359	435	424	411	496	460	O	485	448	323	337	353	370	O	379	240	305	404	O	O	O
362	337	353	R	R	R	O	331	388	268	442	444	359	O	387	295	334	412	379	O	O
307	346	371	310	420	346	O	398	316	324	473	337	362	O	311	412	368	234	389	303	O
280	306	489	478	356	545	420	487	278	502	278	324	307	O	342	431	400	292	448	400	O
404	325	452	O	O	O	O	O	O	O	O	485	280	465	473	458	389	347	455	359	O
428	392	392	430	421	392	354	O	O	O	O	O	O	O	387	468	368	448	296	404	O
448	430	477	418	217	430	O	384	R	R	R	R	382	428	311	302	414	368	302	412	O
336	421	504	289	453	421	O	331	442	497	353	346	437	O	342	430	245	378	431	342	O
347	394	578	325	612	394	O	O	O	310	460	448	322	487	473	342	522	414	299	305	O
304	413	O	O	O	413	O	444	265	380	346	442	265	O	289	429	367	502	497	471	O
394	413	368	323	317	368	O	362	329	O	O	O	O	O	O	396	473	R	R	R	R
O	O	O	263	428	323	O	382	460	454	331	398	485	O	299	522	300	389	310	349	O
R	O	O	382	448	317	O	342	430	289	444	487	485	O	400	388	396	448	282	343	O
342	396	388	256	336	396	O	473	368	395	362	384	331	O	289	337	O	O	321	472	O
398	290	368	281	347	408	O	448	473	315	R	R	R	R	R	R	R	R	R	359	O
394	296	449	290	304	290	O	322	484	449	341	311	506	O	240	301	336	395	349	460	O
371	457	440	341	345	474	O	466	299	300	307	444	437	O	437	379	387	315	437	430	O
310	387	R	R	R	277	O	394	342	342	280	302	412	O	431	371	466	R	R	325	O
O	369	408	382	395	468	O	343	O	O	O	O	299	O	346	430	322	268	310	444	O
R	337	O	O	O	307	O	389	337	300	442	346	468	O	449	460	380	324	380	303	O
O	O	O	O	O	O	O	342	487	442	346	300	400	O	359	449	323	502	454	484	O
O	O	O	O	O	O	O	O	R	R	R	R	316	458	337	449	300	497	289	342	316

	W10							W11							W12						
Truck	D1	D2	D3	D4	D5	D6	D7	D1	D2	D3	D4	D5	D6	D7	D1	D2	D3	D4	D5	D6	D7
T1	321	415	330	O	O	O	O	O	392	229	335	463	366	O	312	387	382	373	387	466	O
T2	456	390	361	O	O	O	O	O	308	R	R	473	500	456	337	500	414	240	522	278	O
T3	400	473	308	O	O	O	O	O	361	458	448	472	394	315	424	518	473	456	388	442	O
T4	377	490	311	490	369	430	354	398	428	388	440	369	O	O	379	430	369	322	337	265	O
T5	482	424	413	496	343	305	395	395	335	500	342	362	R	O	278	411	460	380	299	329	O
T6	246	427	436	484	389	343	315	385	278	368	359	409	O	O	234	379	449	382	342	460	O
T7	450	354	234	354	456	390	O	379	431	425	449	240	321	487	292	337	298	297	473	430	O
T8	420	390	368	390	400	473	O	306	340	319	398	367	490	O	O	O	293	366	448	368	348
T9	490	496	484	496	377	490	O	370	215	396	342	394	371	O	O	O	O	O	O	O	O
T10	369	343	389	343	482	424	O	315	414	295	382	395	375	O	234	354	395	315	412	245	O
T11	430	305	343	305	246	427	O	301	473	299	428	353	323	O	368	390	297	385	407	522	O
T12	354	395	R	R	R	R	O	502	369	362	412	444	487	O	484	496	370	278	396	367	O
T13	398	395	385	450	O	361	O	392	460	311	407	348	321	O	389	343	384	368	427	473	O
T14	398	428	388	418	368	308	O	424	449	413	413	471	353	O	343	305	361	325	328	300	O
T15	395	335	500	518	484	311	O	430	298	436	331	472	O	O	O	395	392	368	364	396	O
T16	385	278	368	325	389	413	O	487	293	234	398	R	R	R	337	395	334	302	464	347	O
T17	395	325	335	430	343	436	O	349	395	368	444	331	398	O	487	335	440	413	342	448	O
T18	395	278	500	354	315	234	O	305	297	484	413	O	O	O	O	395	435	362	394	368	O
T19	O	379	518	398	385	368	O	368	370	389	324	485	395	O	324	289	414	444	353	378	O
T20	O	O	O	428	278	325	O	398	384	343	384	O	342	O	346	289	R	R	R	414	506
T21	430	424	379	388	485	368	O	382	361	315	404	O	O	293	306	331	404	322	240	396	O
T22	411	379	440	418	289	302	O	394	379	385	449	318	324	O	321	398	412	466	437	246	O
T23	337	278	421	379	413	392	O	431	337	278	297	306	366	O	394	382	342	394	431	233	O
T24	O	O	O	O	O	O	O	O	O	368	301	362	419	O	302	321	305	343	346	429	O
T25	449	430	431	431	340	215	414	473	369	325	398	307	303	O	O	O	O	389	449	390	288
T26	335	280	394	484	425	319	396	295	299	368	395	280	400	O	419	448	460	415	316	396	O
T27	334	289	431	394	297	368	O	263	370	302	418	289	341	O	388	349	449	O	O	290	O
T28	433	487	431	484	389	302	O	382	359	392	428	337	342	O	370	343	389	460	471	334	O
T29	R	R	340	425	346	392	O	256	389	334	448	346	396	O	460	472	448	430	370	379	O
T30	O	O	215	319	398	334	O	281	430	440	336	382	412	O	396	359	455	325	343	335	O
T31	472	394	301	428	424	440	O	R	428	435	347	R	349	O	341	O	O	O	O	334	358
T32	369	484	430	424	310	435	O	278	335	430	304	388	371	O	290	387	444	296	487	433	O
T33	362	425	382	403	428	430	O	435	392	319	290	354	310	O	379	430	389	430	428	430	O
T34	409	319	395	392	362	319	O	337	430	394	341	388	420	O	371	382	346	R	R	R	R
T35	240	396	435	R	R	394	O	392	421	485	394	431	444	O	430	395	398	424	379	428	O
T36	367	295	431	484	430	485	O	361	394	331	444	394	463	O	278	334	424	403	367	388	O
T37	394	299	340	425	379	331	O	424	413	362	302	431	376	O	379	420	310	392	435	418	O
T38	334	369	215	319	367	362	O	396	368	325	R	R	R	O	449	335	428	307	341	379	O
T39	420	448	414	396	O	O	O	408	323	296	371	310	420	O	280	448	362	395	369	325	R
T40	335	301	473	295	299	O	O	290	317	411	394	444	302	O	289	301	297	337	342	O	O

The following table illustrates the concluding remarks from the above activity patterns for each truck in the system.

Total No. Days	No. Days Off work	No. Days Repair	No. Days Working	TOTAL min	No. Min Off work	% R
84	20	6	58	22727	2880	12.67215
84	14	5	65	24201	2400	9.916946
84	19	8	57	22855	3840	16.80158
84	16	6	62	22752	2880	12.65823
84	14	1	69	26076	480	1.840773
84	17	3	64	23750	1440	6.063158
84	11	3	70	26187	1440	5.498912
84	13	4	67	25504	1920	7.528231
84	26	3	55	20663	1440	6.968978
84	10	6	68	25419	2880	11.33011
84	15	0	69	25748	0	0
84	10	7	67	26037	3360	12.90471
84	19	0	65	24264	0	0
84	16	0	68	25709	0	0
84	21	0	63	24513	0	0
84	14	6	64	24280	2880	11.86161
84	21	0	63	24066	0	0
84	27	4	53	20221	1920	9.495079
84	16	0	68	26035	0	0
84	17	6	61	22359	2880	12.88072
84	13	1	70	25694	480	1.86814
84	9	7	68	26014	3360	12.91612
84	17	1	66	25446	480	1.886348
84	26	0	58	21399	0	0
84	13	4	67	25826	1920	7.434368
84	16	0	68	25788	0	0
84	13	4	67	24932	1920	7.700947
84	16	0	68	26112	0	0
84	22	6	56	20700	2880	13.91304
84	20	1	63	23341	480	2.056467
84	16	6	62	22915	2880	12.56819
84	16	0	68	25230	0	0
84	11	9	64	23989	4320	18.00825
84	20	6	58	21212	2880	13.57722
84	14	2	68	26697	960	3.59591
84	17	7	60	23041	3360	14.5827
84	18	0	66	24948	0	0
84	15	6	63	23470	2880	12.27098
84	25	1	58	21561	480	2.226242
84	20	6	58	20832	2880	13.82488
Average	16.83	3.38	63.80	24062.83	1620.00	6.92

F.14.1 Trucks breakdown pattern

Month	Number of trucks with no breakdown during the month				
	1383	1382	1381	1380	1379
1	14	12	11	12	10
2	10	12	12	11	12
3	12	8	10	8	11
4	11	10	14	7	10
5	8	12	12	14	7
6	10	11	10	14	12
7	12	7	11	10	14
8	11	14	12	11	14
9	12	11	8	12	12
10	-	10	10	11	11
11	-	11	11	12	11
12	-	13	12	10	10
Average	11.11	10.92	11.08	11.00	11.17
Over all Average		11.06			