

Old Dominion University

ODU Digital Commons

Electrical & Computer Engineering Theses & Dissertations

Electrical & Computer Engineering

Summer 2001

Simulation Optimization Studies of Routing and Process Flow Problems

Satish Ravichandran
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/ece_etds



Part of the [Electrical and Computer Engineering Commons](#), and the [Systems Engineering Commons](#)

Recommended Citation

Ravichandran, Satish. "Simulation Optimization Studies of Routing and Process Flow Problems" (2001). Master of Science (MS), Thesis, Electrical & Computer Engineering, Old Dominion University, DOI: 10.25777/8ste-bf59
https://digitalcommons.odu.edu/ece_etds/166

This Thesis is brought to you for free and open access by the Electrical & Computer Engineering at ODU Digital Commons. It has been accepted for inclusion in Electrical & Computer Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

SIMULATION OPTIMIZATION STUDIES OF ROUTING AND PROCESS FLOW PROBLEMS

by

Satish Ravichandran
B.E., Bharathiar University, Coimbatore, India

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
August 2001

Approved by:

Dr. Ravindra P. Joshi (Director)

Dr. Linda L. Vahala

Dr. Vijayan K. Asari

ABSTRACT

SIMULATION OPTIMIZATION STUDIES OF ROUTING AND PROCESS FLOW PROBLEMS

Satish Ravichandran
Old Dominion University, 2001
Director: Dr. Ravindra P. Joshi

Computer aided simulation is emerging as a powerful tool for numerical analysis and in conducting performance evaluations of complex systems that depend on a multitude of variables. The primary objective in such simulation studies is to gauge the performance of the system under a various constraints and operating conditions. The effects of changing the operating parameter space can thus be analyzed without having to implement costly changes. Simulations are also carried out for the baseline scenarios to verify and validate the basic underlying system model. In this thesis research, two practical problems were studied through numerical modeling, and optimized solutions obtained for both. Optimizing the pick-up and delivery routes using a commercial software tool was the first task. Optimization of a production assembly line using a discrete event simulation tool was the second project that was carried out. The primary objective for the first task was to explore various routing scenarios and determine delivery routes that would minimize the total network mileage, while maintaining the pick-up time slots requested by the clients. A related task was to evaluate the possible advantages of centralizing all routing activity from a single site, instead of the two-hub scenario currently in effect. A total of eight different scenarios were studied as part of this effort. The second task involved optimization of the throughput of a fuel injector plant by placing buffers within the assembly lines for increased productivity.

TABLE OF CONTENTS

	Page
LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
Chapter 1. INTRODUCTION	
1.1. MODELING AND SIMULATION: AN OVERVIEW.....	1
1.2. ROUTING AND SIMULATION.....	2
1.3. DISCRETE EVENT SIMULATION.....	3
1.4. ORGANIZATION OF THE THESIS.....	4
Chapter 2. BACKGROUND AND LITERATURE REVIEW	
2.1. INTRODUCTION.....	5
2.2. GENERAL CLASSIFICATION OF ROUTING PROBLEMS.	7
2.3. SOLUTION STRATEGIES.....	10
2.3. DATA REQUIRED FOR ROUTING PROBLEM.....	14
2.4. CONSTRAINT SPECIFICATION.....	18
2.5. STUDY OF PROCESS FLOW IN A MANUFACTURING SYSTEM.....	20
2.6. RESEARCH ISSUES FOR THE PROCESS LINE PROBLEM.....	26
Chapter 3. DETAILS OF THE SIMULATION TOOLS USED IN MODELING	
3.1. INTRODUCTION.....	28
3.2. FEATURES OF VARIOUS ROUTING PACKAGES.....	30
3.3. ADVANTAGES AND BENEFITS OF SIMULATION AND MODELING.....	32
3.4. ROUTEPRO – SIMULATION MODEL.....	34
3.5. ROUTING ALGORITHMS.....	39
3.6. REVIEW OF ARENA	
3.6.1. CONFIGURATION OF MENUS.....	49
3.6.2. CONFIGURATION OF TOOLBARS AND TEMPLATES.....	52
Chapter 4. RESULTS AND DISCUSSIONS	
4.1. INTRODUCTION.....	56
4.2. PROBLEM #1: INTRODUCTION AND BACKGROUND	
4.2.1. DATA SHEETS AND DIALOG BOXES IN ROUTEPRO.....	59
4.2.2. INITIAL TESTING FOR THE ROUTING ANALYSIS.....	60
4.2.3. SIMULATION RESULTS.....	61
4.3. PROBLEM #2: PROCESS FLOW OPTIMIZATION	
4.3.1. PLANT LAYOUT.....	72

4.3.2. DETAILS ON THE SYSTEM DATA..... ..	75
4.3.3. SIMULATION MODEL.....	77
4.3.4. SIMULATION RESULTS.....	82
Chapter 5. CONCLUSIONS	
5.1. INTRODUCTION.....	86
5.2. RESULT SUMMARY FOR ROUTING PROBLEM.....	87
5.3. RESULTS OF THE PRODUCTION LINE PROBLEM....	90
5.4. FUTURE WORK.....	93
REFERENCES.....	96
VITA	101

LIST OF TABLES

Table	Page
4.1. Baseline Analyses.....	62
4.2. Routepro optimized two hub results.....	64
4.3. Max 3 –hour run with single Depaul Hub (all time constraints met).....	66
4.4. Max 4 –hour run with single Depaul Hub (all time constraints met).....	67
4.5. Max 3 –hour run with single Depaul Hub (12 of 198 time windows not met)...	69
4.6. Max 3 –hour run with single Depaul Hub (minimal constraints).....	71
4.7. Uptimes and Downtimes of various machines.....	81
5.1. Summary of the results of the routing problem.....	89

LIST OF FIGURES

Figure	Page
2.1. Simple Serial Production line.....	21
3.1. Greedy Algorithm for one weight function.....	45
3.2. Basic building block of any model.....	54
4.1. Plant Layout.....	74
4.2. Block Diagram of the model with efficiency.....	80

CHAPTER 1

INTRODUCTION

1.1 MODELING AND SIMUALTION: AN OVERVIEW

Vehicle routing problems have been a topic of research for a very long time. Vehicle routing problems involve the navigation of one or more vehicles through a network of locations with each location being serviced a predetermined number of times. In vehicle routing problems, a set of vehicles with limits on capacity and travel time are available to service a set of customers with demands. Specifications on the earliest and latest service times are also generally given. The objective is to minimize the number of vehicles and the distance traveled for servicing the set of customers without being tardy or exceeding the capacity or travel time of the vehicles. With the rapid advancement in the technology, there been an urge to improve the quality of the service and also to attain faster service. Several algorithms have been developed to bring about an effective and efficient method for optimizing the routes that are involved in the service. Many simulation tools have also been developed with all of the algorithms to enable the simulation of actual routing scenario. One of the topics of this project involves the application of one simulation software tool to solve a routing problem faced by a chain of hospitals for the pick up and delivery of the sample specimens from doctor's sites and laboratories.

Another area of interest and the topic of the thesis is the optimization of the production line in a manufacturing plant. Similar to the previous problem, another simulation tool is used to model the manufacturing plant and find an optimal solution. The primary

References in "International Journal of Production Research" format.

objective of this second project was to increase the overall efficiency of the plant and also to improve the throughput at a minimized cost of production.

1.2 ROUTING AND SIMULATION

Simulation can be defined as the imitation of a dynamic system using a computer model to improve the performance of the existing system. There have been great advances in the field of the microprocessor design; thus the speed of the processor has been improving every year. This has helped the field of computer simulations by focusing more attention on simulation work to attain more feasible and optimal solutions. Such numerical modeling provides useful insights into the behavior of complex systems. Various companies have developed simulation software tools and every software has its own unique features. The basic idea involved in the simulation of the routing system involves the input of the various parameters and specifications for the system such as the desired time windows, capacities, maximum time of travel, etc. Also, every routing software contains a geographical database containing information about the various routes and roads for a particular location. This database is the key to any routing software and has to be updated on a regular basis. The software also contains various algorithms and the one which best suits the model is chosen for arriving at the optimal answer. Using computer-aided simulations is advantageous in this regard, as it facilitates the analysis of a large number of possible routing scenarios. Another advantage of the simulation software is the graphical and visual capability of objects and their visualization effects. Such graphical interfaces for the routes help in providing a better understanding to the user.

This thesis deals with the optimization of the routes for a chain of hospitals, subject to a set of given constraints. These constraints and requirements included operating within desired time windows, obtaining solutions that reduced the total travel distance, and minimizing the number of vehicles used.

1.3 DISCRETE EVENT SIMULATION

Discrete event simulations are generally used to replicate systems for which events take place at discrete but random time instants. The inputs to such models are given using a probability distribution functions from which the results are generated. This helps to make the model more realistic. It also allows for more simulations and to explore a variety of possible scenarios for relative evaluation. The random inputs introduce a corresponding stochastic element to the output as well. One of the examples of such a system is the production line system in a manufacturing plant. The machines may fail at random instances and thus has an affect on the output of the plant. Also, in most of the manufacturing plants, discrete parts are being manufactured; hence, discrete event simulations are a natural approach.

The second part of this thesis deals with such a manufacturing problem that involves a discrete event simulation model. The objective of the thesis is to model and simulate a manufacturing plant the produces fuel injectors for cars. The plant is modeled using commercial simulation software, and the effects of adding buffers between machine units studied to evaluate potential operating efficiencies. A suggestion on optimizing the buffer in the production line is also done by carrying out different simulation runs at each position.

1.4 ORGANIZATION OF THE THESIS

This thesis focuses on the use of simulation tools for modeling and optimization of practical systems. The thesis is organized into five chapters. The first chapter gives an introduction to the thesis, the routing problem and discrete event simulation of a production line. The first chapter also gives an overview of the subsequent chapters. The second chapter deals with the literature review and the background. This involves the various techniques used to solve the routing problem, the heuristic methods and the literature issues behind the process flow systems. The third chapter discusses the various simulation tools available and the reason behind the selection of the particular software for this thesis research. The advantages and limitations of the software and the features of the software are discussed. The chapter also discusses the various algorithms used in solving the routing problem. Details of a method to calculate the efficiency of the production line quantitatively are also presented. The fourth chapter deals with the results of the vehicle routing and the production line optimization problems. The implications of the results obtained are discussed in depth. The fifth chapter presents the conclusions of the two problems that were defined and formulated in the fourth chapter. Finally, this chapter concludes with a section on possible future work in this area.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

2.1 INTRODUCTION

One of the tasks of this research is to determine an optimal routing solution for the delivery and pick-up for a health-care organization. Consequently, it is important to first discuss and define routing problems and their solution techniques. This is done in this chapter, and an appropriate literature review is included. Basically, all vehicle routing problems can be categorized in the following general terms. Given a set of customer locations and a set of vehicles of specified size and capacities, one has to devise a routing strategy to visit each of the locations subject to certain given constraints [1]. The objective is to minimize a function of a given variable (e.g., the total distance traveled or the round trip time) and is usually subject to some constraints that are specified (e.g., the capacity of the vehicles, or delivery time slots). There are different ways to solve this problem and routing designers are continually coming up with faster and better solutions. Since the vehicle routing problem can have many variations, there are generally a number of approaches for a solution. In order to properly select the most appropriate technique, it becomes necessary to first formulate and construct a suitable model for subsequent solution. After the model formulation, an optimal solution typically needs to be found using either exact methods or heuristics depending on the complexity of the model.

A second objective of this research is to analyze a particular production flow problem, and provide numerical solutions for improving the in-house processing throughput and

optimizing the production system for overall efficiency. In so doing, the solution should provide enhanced yields, profitability and growth for the organization.

Before embarking on any system modifications or design changes, it is desirable to know the consequences of such variations and moves. This desire to ascertain the potential after-effects a priori is the basic driving force behind numerical simulations. Computer simulation refers to methods for evaluating a wide variety of real world systems through models and their software implementation designed to imitate the system's operations or characteristics. Simulations allow for the analysis of a wide range of scenarios. The primary objective of this project is to estimate the efficiency and potential productivity enhancement by introducing a buffer stage in the DEKA assembly line for production. The DEKA system is a proprietary production line for processing automobile injectors at a local Siemens plant. In theory, the throughput of a manufacturing system always increases when buffers are introduced for temporary storage. This occurs because buffers reduce or eliminate the "starvation" and "blockage" times as discussed later in this thesis. However, the introduction of buffers comes at a price. There are actual costs for the physical buffer structure, and in addition, the intangible costs of creating the space required for accommodating the buffer. The latter usually displaces other working units for which additional space and connectivity needs to be furnished. Due to this trade-off between buffer size and capital cost, there is an optimal level of buffer capacity that is desirable. One of the objectives in this work is to obtain such an optimal size for the DEKA assembly line buffer through stochastic simulations of the production flow process.

2.2 GENERAL CLASSIFICATION OF ROUTING PROBLEMS

The essence of a vehicle routing problem is the allocation and deployment of a fleet of vehicles in such a way that the solution reduces the cost of transportation, the travel time and also, if possible, the number of vehicles used. All the above solutions have to be achieved without any compromise on the quality of service. The quality of service in the context of the present project implies that the delivery of samples and reports to and from various hospitals and doctors' sites should be occur within specific time windows. There is to be no compromise on changing these time windows. In general, almost all of the present day routing scenarios can be classified into the following routing problems.

(i) The Traveling Salesman Problem (TSP): This is one of the oldest of routing problem and many other routing problems are a variant of TSP [2]. Therefore, this continues to be the benchmark and also many newer algorithms are derived from this. This problem is about a salesman who travels many cities to sell his products. It involves the task of finding the shortest route possible covering all the cities. The salesman has to visit each city at least once during his trip. The problem may be symmetric or asymmetric. In a symmetric traveling salesman problem, the cost of the travel is the same whichever direction of travel he takes. But in asymmetrical TSP the direction, which the salesman takes adds to the cost of the trip.

(ii) Chinese Postman Problem (CPP): This problem deals with the issue of visiting a lot of places with the shortest distance possible and also, with minimum cost [3]. The traveler has to visit all the possible places at least once. As in traveling salesman

problem, the routes can be directed or undirected. The solution for this problem is obtained using polynomial bounded algorithms. The mixed CPP which has a mixture of directed and undirected routes is NP-hard [4].

(iii) Multiple Traveling Salesman Problem [5]: This is a variant of the TSP where there is more than one salesman (or vehicle). All the vehicles need to leave from and return to a one common depot. The condition is that each vehicle must visit at least one node.

(iv) Single Depot Multiple Vehicle Node Routing Problem [6]: This problem has one depot and multiple vehicles. Its objective is to establish or determine routes, which service all customers covering a minimum distance. It differs from the multiple TSP in that the capacity of the vehicles is limited and the demand at each node is deterministic.

(v) Multiple Depot, Multiple Vehicle Node Routing Problem [6]: This is a variant of the above in that in this case there are many depots instead of a single central depot. The vehicles have to service all of the depots. But the vehicles must leave and return to the same depot. A small variant to it is the problem with stochastic demands in which the demands of the customers are uncertain.

The above general categories of routing problems are the ones on which most of the algorithms are based. However, there are a few other classes of routing problems, which are generally encountered, in daily life. They are the simple vehicle routing problem

(VRP), the vehicle routing problem with time deadlines (VRPTD) and vehicle routing with time windows (VRPTW).

(vi) Simple Vehicle Routing Problem (VRP) [7]: This is the most common form of routing problems. The problem requires the determination of the routes servicing all customers in a particular geographic region with the minimum cost of operation. There can be one depot or many depots. This routing problem does not have any constraints.

(vii) Vehicle Routing Problem with Time Deadlines (VRPTD) [7]: VRPTD is a routing problem which requires the vehicles to service customers within a deadline. So, time becomes the major constraint. VRPTD is similar to VRPTW, which is discussed next, except for the fact that it does not have an upper bound on time. This kind of problem is noticed in our day-to-day lives. For example, routing of school buses in a city requires the children to be dropped in their respective schools before a particular time, which then becomes the time deadline for that school. Similarly, there are many industries where the delivery of goods or raw materials needs to be carried out before a time deadline. Since customer service is one of the most important things careful and precise routing assumes at most priority. There are various algorithms specific to VRPTD.

(viii) Vehicle Routing with Time Windows (VRPTW) [7]: The VRPTW requires the routing of vehicles in such a way that they reach the customer locations within a particular interval of time known as time windows. Time windows arise because of

business organizations that work on fixed time schedules. Specific examples include bank deliveries, postal deliveries and industrial material deliveries. Simple routing problems include only the cost of travel distance and time cost but the presence of time windows include the cost of waiting time incurred when a vehicle arrives too early at a customer location and has to wait for some time. Similarly, if a vehicle arrives late, the customers are unhappy and there is a cost associated with the displeasure. This problem has been given importance recently and specific algorithms and heuristics have been formulated to tackle this particular scenario.

2.3 SOLUTION STRATEGIES

The routing model determines the method of solution for the vehicle routing problem. In the case of a small model with a few nodes, the problem can be solved with an exact mathematical representation. For larger models one resorts to commercial packages available in the market. These packages are very convenient to use, are user-friendly, and have good graphics and animation capabilities. They perform quite large routing problems using intrinsic algorithms. The computers of today make them a very attractive and viable solution. However, there are some routing problems that can be complicated. The software packages also find them difficult to handle. Heuristic methods are often used to obtain optimal solutions for such complicated problems. A common heuristic method for solving the vehicle routing problem is presented below.

The Nearest Neighbor Heuristic: This heuristic method starts every route by locating the unrouted customer closest to the depot [8]. For each iteration, the heuristic searches for the closest customer to the last customer appended on the route. Customers are added

depending on feasibility issues like time windows and capacity restrictions. A new route is initialized, when no customer is left to satisfy the constraints for the route under consideration a function is used here compute the geographical and temporal closeness of the customers.

Let “i” be a customer who was routed last on a particular route and let “j” be the customer who could be visited next. The function C_{ij} estimates the straight distance between two customers, the time difference between the completion of service at i and the beginning of service at j, and the urgency of delivery to the customer j. Based on this customers are routed. In this context, one sets up and formulates an appropriate function C_{ij} .

The Tabu Search Algorithm: The Tabu search is a local search heuristic [9], which explores the solution space by moving at each iteration from a solution “s” to the best solution in a subset of its neighborhood $N(s)$. To avoid cycling, solutions possessing some attributes of recently explored solutions are temporarily defined as Tabu or forbidden, unless their cost is less than a defined aspiration level. The algorithm uses a heuristic, which does insertions of unrouted customers or removals of customers from their current routes and their re-insertions into different routes. Beginning with three arbitrary vertices, this heuristic constructs a Hamiltonian route by inserting at every step a vertex v between its nearest neighbors v_i and v_j on the current route. The algorithm does least cost insertion to insert the vertex v . The reverse case is utilized to delete a customer from a route.

Insertion Heuristics: An insertion procedure does a sub-tour on k nodes starting at one of the nodes and tries to understand which node should enter the sub-tour next and then

understands where the new node must be inserted [10]. The tour starts by selecting a node i . Then node k is found which enables the computation of C_{ik} , where C_{ik} is a function that depends on the type of insertion method used. This addition forms the sub-tour $i-k-i$. Given a sub-tour, one then finds node “ t ” such that it is closest to any node in the current sub-tour. After finding node t , the arc (i, j) in the sub-tour which minimizes $C_{it} + C_{tj} - C_{ij}$ is determined. Finally, this new node t is inserted in between the two nodes i and j . A kind of insertion heuristic is the Push-Forward Insertion Heuristic (PFIH), the PFIH starts a new route by selecting the first customer and then inserting customers into the present route until either the capacity of the vehicle is exceeded, or it is not feasible to add another customer to the emerging route. The means for selecting the first customer is based on the cost of the customer farthest away from the depot with the earliest time deadline and the minimal polar coordinate angle. It is presumed that each customer has a deadline. The arrival time of a vehicle at a customer site must be before its deadline. For a customer C_i to be inserted between the depot and customer C_1 , computing the duration by which the arrival time of t_1 is pushed forward checks the insertion feasibility. A change in the arrival time for t_1 could affect the arrival times of all the successive customers of t_1 in the current route. The insertion feasibility for C_i is computed by sequentially checking all the successor customers of C_i for feasibility. If the insertion of C_i into the route does not affect the arrival time of t_j at C_j , then the push-Forward for a customer is zero. The sequential checking for feasibility is continued until the push-forward for a customer is zero, or all customers are checked for feasibility. The priority is in the order of distance, polar coordinate angle and latest time. When calculating the polar coordinate angle of the customer with respect to the depot, the angular value of the

customer from the depot is normalized in terms of the distance as was done in the case of the “Deadline Sweep Heuristic” discussed in the next section. Once the first customer is selected for a route, the heuristic selects an unrouted customer C_l and an edge $\{i, j\}$ in the current tour that minimizes the total travel cost and inserts customer C_l between i and j if it does not violate time feasibility and capacity constraints. A new route is started if no more customers can be inserted without overriding the capacity or time feasibility constraints.

Sweep Heuristics: The sweep heuristic is effective approach for about three hundred nodes [11]. The solution is based on two steps. Primarily, it assigns nodes to vehicles and then secondarily sequences the order in which the vehicle traverses the nodes. A seed node is randomly selected. With the pivotal central depot, it starts sweeping the ray from the central depot to the seed. Nodes are added and clustered in this process. The polar coordinate indicating angle for the nodes are ordered from the smallest, i.e., zero, to the largest. The seed’s angle is zero. Routes are magnified as the angle increases until the capacity restrictions stop the sweep. Then the next demand node is set as the seed and the sweep starts.

Genetic Sectoring Heuristic (GSH): An adaptive heuristic search method based on the genetic search algorithm is the genetic search heuristic [12]. The GSH is a cluster-first route-second method. The genetic algorithm is an iterative process that stabilizes a population of P candidate members over multiple simulated generations. String entities of chromosomes form the population. The chromosomes are bit arrays or strings. Binary values (alleles) are placed at each position (or locus) of the array. Each chromosome has a fitness value linked to it. The chromosomes from one generation to the other are

decided with emphasis on the fitness value. Similarly the GSH clumps the customers using the genetic algorithm and then they are routed using the already defined insertion method. The customers are clustered in sectors in this process. They are split into V sectors where V is the number of vehicles. Hence each customer is related to one of the V vehicles and each sector has its related vehicle. After the customers are sectorized using a sectoring process, the genetic algorithm is used to select the sectors that will minimize total cost. Then the customers linked to each sector are routed using the most economical insertion method. This forms the skeletal basis of the genetic sectoring heuristic.

Many more methods and algorithms which solve other variations of the routing problem are available. Those briefly explained above show the routing heuristics that have been employed for some common routing problems. Archived materials are available for the routing problems. Depending upon the problem one chooses either exact methods or routing based on heuristics. However, with faster computers and the need for an easier approach to routing problems a new idea of dealing with the routing problems is sought. The routing software packages are convenient to use, offer good animation, quite flexible and most important of all, quite accurate. In this simulation RoutePro developed by CAPS Logistics, Inc., will be used and applied to the situation.

2.4 DATA REQUIRED FOR ROUTING PROBLEMS

Data is the basic building block for any model. In order to tackle a routing problem, one has to have all the data before proceeding with the computer simulation. The basic data required for the vehicle routing problem are

- (i) Fleet setup,

- (ii) Pick-up and delivery time windows,
- (iii) Geographical database of the location,
- (iv) Demands from the customer.

(i) Fleet setup:

Vehicles are an important parameter in any routing problem as they are the primary means of transportation. The cost associated with the fleet involves the cost of the vehicle, maintenance cost, fuel cost, etc. Among these, the cost of the vehicle accounts for the maximum and so a proper selection of fleet becomes necessary. The selection of vehicles varies depending on the type of pick up and delivery the firm is operating. The pick up and delivery can be broadly classified under the following classes:

- (a) Direct pickup and direct delivery: Under this classification, the same truck used for pickup and delivery also performs the long distance route. The load, therefore, need not change trucks between its pickup and delivery. This is the case when there are no intermediate stops between the pickup and the delivery points. The selection of the vehicle here depends on the load. Generally in these cases, the loads are large, necessitating large carrier trucks. The best example for this is the case of garbage disposal. A large vehicle picks up all the garbage from the pickup points and then directly takes them to the dump yard. There is no intermediate stage.
- (b) Indirect pickup, direct delivery: In this case, the load is picked up at the pickup point from a carrier vehicle and then sent to its corresponding “sending depot.” The sending

depot then delivers the load to its destination directly. In this case, the pickup points have slightly smaller loads. So smaller vehicles like minivans can collect the loads from these pickup points and take them to the sending depot. At the sending depot, all these smaller loads are collected and sent to the destination directly by a larger vehicle.

- (c) Direct pickup, indirect delivery: Here the same truck performing the long distance hauling directly picks up the load and after performing the long distance tour delivers the load to its corresponding receiving depot. From this depot, a separate carrier vehicle takes delivery of the partial loads. An example for this is the delivery of milk to various stores from the milk production plant. The milk cans are sent in large loads through a large truck to a receiving depot. This depot sorts them out according to the demand of various stores and sends them in smaller vehicles.
- (d) Indirect pickup and indirect delivery: In this mode, the pickup and the delivery are indirect. The partial loads are picked up from various pickup points and are sent to the corresponding sending depot. They are then hauled onto larger vehicles and are sent to the receiving depot. Next, the loads are then delivered to various delivery points. So partial loads have to be handled twice in depots. A very good example for this mode is that of the postal system. The mail packages are picked up from various pickup points and are sent to the main post office. There they are all sent to the receiving depots through larger vehicles. The receiving point is also a central post

office. It then sends out all the pieces of mail to the delivery points by smaller vehicles. Commercial airliners also come under this mode of transport.

Once the mode of pick up and delivery is known other data such as the capacity of the vehicle, mileage of the vehicle, maintenance cost, and the average speed that the vehicles can travel. This implies that detailed information is necessary in setting up the various parameters for the fleet setup.

(ii) Pick up and delivery details:

The pick up and delivery data contains information regarding address of the each pick up and delivery location. This data also gives an idea of the current routing the company is adopting and this data becomes very useful in conducting the base run simulation. Thus, the software's efficiency is verified using this data. The software also needs information like the quantity of the data that has to be delivered at each location, the time window that the pickup and the delivery sites are available for operation.

(iii) Geographical details:

The geographical details consist of the road networks and information regarding the general layout of a geographic area. The geographical database also consists of the speed limits at each road and the type of roads (interstate, residential etc.). The routing practitioner should have updated data about the roads, intersections, speed limits, etc. Customer locations are also considered in this category. Another important feature in using the geographical database is that the route from one location to another can be either a direct network or a road network. In direct network, line-of-sight routes are

followed and so it does not give the actual distance. In the road network, the simulation takes the existing roads and follows that path to go from one location to another. This is the reason that the road network database has to be updated.

(iv) Demands:

In any routing problem, the customer's nature of demand is of primary importance since it determines other factors like routes and vehicle capacities. The quality of service should not be compromised at any cost. The demands are of three types: (a) Deterministic Demands [13], (b) Stochastic Demands [13], and (c) Real Time Inflow Demands. Of these, the deterministic demands are known well in advance. With this kind of demand, the routing practitioner is in a better position to route customer locations and assign vehicles to them. Stochastic demands, as the name implies, are uncertain or not fully known a priori. This makes it very difficult to route and to assign proper size vehicles since the revenue from these types of customers is probabilistic in nature. Real time inflow demand is also a kind of uncertain demand, but the demand is known just before the vehicles embark upon their tour. This requires on-the-spot decision-making capacity.

2.5 CONSTRAINT SPECIFICATIONS

As mentioned previously, every routing problem has many variations and the constraints for each problem also varies. The constraints can be classified as follows: (i) Route constraints, (ii) Vehicle Load constraints and (iii) Geographical constraints. Details on these various constraints are given next.

(i) Route constraints:

Route constraints involve constraints like speed restrictions, bad roads, intermediate breaks for the drivers, traffic intensity and traffic lights. All these factors impose restrictions on the route time. Overnight trips also place restrictions since the driver cannot drive the vehicle for not more than few continuous hours.

(ii) Vehicle load constraints:

The vehicle load constraints mainly depend on the amount or the capacity of the load that a vehicle can carry. This amount of the load may involve weight or the volume of the parcel that has to be transferred. When the customer demands are quite large compared to the maximum load capacity of the vehicle, multiple trips have to be made to satisfy the customer. This increases the cost of the transportation.

(iii) Geographical constraints:

The geographical constraints are very similar to the route constraints except that the geographical constraints also involve place restrictions. For example, some goods carried in one state may not be permitted in other state without surcharges. Thus, this cost adds to the transportation cost. Additional charges such as the toll fee, tax etc., are also levied for using the particular road. Also, the speed limit in a particular road plays a very important role in maintaining the time of delivery.

All these constraints are kept in mind before formulating any route. This study becomes a very critical factor and plays an important role used in optimizing the vehicle routing problem.

2.6 STUDY OF PROCESS FLOW IN A MANUFACTURING SYSTEM

As stated in this chapter's introduction, a second problem being studied in this thesis relates to the optimization of a manufacturing flow process. The production line in any manufacturing system consists of a series of workstations that perform different operations. Jobs flow through the line and are processed at each workstation [14,15]. The throughput of the production is defined as the number of jobs that can be processed (or articles that can be manufactured) by a number of machines along the production line. The machines in a simple production line have a constant processing time and the transfer jobs between stations are synchronized. However, in an advanced production line, each machine operates at different processing times and the machines can be in parallel. This can result in a higher throughput of the production line. The problem encountered in justifying the installation of an automatic transfer line is the low throughput of the line. One way to improve the throughput, and also one of the objectives of this project, is the introduction of buffer in between each station. Figure 1 shows a simple serial production line with a buffer in between each station.

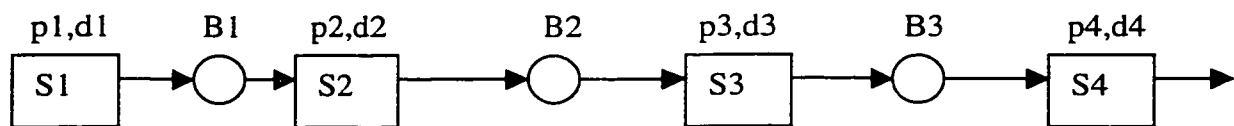


Fig. 2.1. Simple serial production line.

In the above figure, the following notation has been used:

S_1, S_2, \dots are the different stations in the production line,

B_1, B_2, \dots are the buffer sizes in between two stations,

d_1, d_2, \dots are the downtimes of each station, and

p_1, p_2, \dots are the processing times for each station, i.e., the time taken to process a job.

The buffers between stations are used to accommodate jobs that have been processed in one station and are waiting to be processed at the next station. When the next station fails to operate, the jobs are added on to the buffer and are used up at a later time when the station gets repaired. Also, buffers are very useful when the stations in the production line have a variable processing time. If all the stations have same processing time, then the jobs just pass through the buffer and no jobs are stored in the buffer. But in case of different processing times, if the next station process the jobs slowly compared to the present station, then the jobs are stored in the buffer and are sent to the slower station later.

If a buffer gets full, then the upstream stations are blocked of new jobs. This is commonly referred to as the “blockage” problem. If a buffer is empty, the downstream stations are “starved” of new jobs. The concept of blocking and starving occurs due to variable processing times and because of frequent downtime of the individual stations. If the interval between successive downtimes, and/or the duration of the downtimes is unequal, then such blocking and starvation problems occur. The downtime of a station is defined as the length of stoppage of a station [16]. Therefore, introduction of buffers between stations increases the throughput to a large extent. With the introduction of

buffer between the stations with infinite storage capacity, the problem of starving and blocking will be maximally reduced. However, the introduction of buffer storage is expensive due to the expense of the physical structure, and the intangible cost of creating the space required for accommodating the buffer. The latter usually displaces other working units for which additional space and connectivity need to be furnished. Due to this trade-off between buffer size and capital cost, there is an optimal level of buffer capacity that is desirable. One of the goals, therefore, is to obtain an optimal size for the assembly line buffers without incurring excessive cost. Several methods of doing this have been discussed in the literature [17-23].

The usual rationale behind studies of production lines is that even slight improvements in the efficiency (or throughput) of one machine of the stochastic system can greatly enhance the overall production. A common measure of efficiency is the utilization of the assembly line, defined as the ratio of the actual system output to the theoretical maximum. In a perfectly balanced and identical deterministic system, the utilization is 100% since transfer of work from one station to the next proceeds in a synchronized fashion. Any imbalance breaks this perfection and lowers the utilization. Such imbalances can arise from stochastic breakdown of machines, differences in processing times of individual units, and inadequate coupling (or feed through) between stations. Incorporating storage spaces between workstations to “buffer” the irregularities can reduce the deleterious impact of coupling inefficiencies.

One of the classic problems of industrial engineering is to accurately predict the efficiency of practical assembly and/or production lines. One of the earliest studies in this field was by Hunt [24] who determined that the maximum attainable utilization U_{\max}

for a two-station system with identical exponentially distributed processing times is given by:

$$U_{\max} = (B+1)/(B+2) , \quad (2.1)$$

with B being the interstation buffer capacity. Many subsequent researchers extended this result. Hillier and Boling developed a computationally intensive but exact procedure for solving short, tightly coupled, unbuffered lines [25]. Basu extended the results from the exponential distribution to other functions [26]. A simple approximate expression for the utilization “ U ” in an N -workstation system having identical buffers of size “ B ” was:

$$U = \{(B+1) N\} / \{(B+3) N - 2\} . \quad (2.2)$$

Later, Muth [22,23] developed expressions using both theoretical analysis and curve-fitting procedures. Blumenfeld [14] later generalized Muth’s formula.

J.A. Buzacott carried out a numerical analysis on the production line throughput with the buffer size as a parameter. The following variables are used to derive a formula for the throughput based on buffer size [16]: (a) *Uptimes*: given by the number of working line cycles between stoppages, (b) *Downtimes*: defined as the length of stoppage of a station, (c) *Time Between Stoppages*: given as the time from the beginning of one stoppage to the beginning of next, and (d) *Buffer stocks*: defined to be the buffer storage sizes. For an N -station system, the number of buffers was $N-1$.

Probability distribution functions were used to denote these time units as the failure of a station occurs at random time and for random time units. Therefore, to model the system very close to realistic probability distribution functions had to be constructed.

In the analysis, three case studies were analyzed to obtain the efficiency of production line. These three cases were: (i) Production lines with no buffer capacity provided. (ii) Systems with infinite capacity buffers between each station, and (iii) Addition of a buffer of constant size B, between each station. The results obtained are summarized below.

(i) Lines with no buffer:

In this case, a station will be forced down whenever there is any break in the line. Let M be the number of stations in the production line, T_i the mean repair time for the i^{th} station, the active processing (or operating) time of the i^{th} machine be defined in terms of a distribution with mean $1/a_i$. Here, a_i is the probability that the station stops functioning or is down. If the stations are up for a time R and a total of S cycles, then the number of times “n” that the entire line is down will be given as:

$$n = n_1 + n_2 + n_3 + \dots + n_m, \quad (2.3)$$

where n_i is the number of breakdowns of the i^{th} station in the time interval R. Therefore, the total time the line is down, T_{down} in time R works out to:

$$T_{\text{down}} = \sum_{i=1}^M n_i T_i = \sum_{i=1}^M a_i T_i S, \quad (2.4)$$

The line efficiency A_o is therefore given as:

$$A_o = S / [S + \sum_{l=1}^M a_l T_l S] = 1 / [1 + \sum_{l=1}^M a_l T_l] \quad . \quad (2.5)$$

If the mean repair time of each station is the same and equals T, then

$$A_o = 1 / \{ 1 + Ta \} \quad . \quad (2.6)$$

(ii) Line with infinite capacity buffers:

With infinite buffer the stations are independent to each other and the effects of station breakdown or the repair is reduced to the minimum. Therefore, the production line efficiency for an infinite buffer is given as:

$$A_\infty = \min \{ A_l \} \quad . \quad (2.7)$$

(iii) Line with constant buffer:

A generalized formula relating the throughput, λ of a production line with a constant buffer B has been derived and given by Dennis E. Blumenfeld [9] for this case to be:

$$\lambda = \frac{1}{T \left[1 + \frac{1.67(N-1)C}{1 + N + 0.31C + 1.67NB/(2C)} \right]} \quad , \quad (2.8)$$

where N is the production line length, and C is the coefficient of variation of processing times.

2.7 RESEARCH ISSUES FOR THE PROCESS LINE PROBLEMS

The above discussions convey several of the general concepts and issues involved in this area. Thus, for example, the following points can easily be surmised. (i) First, it is clear that the throughput can generally be maximized for a balanced, deterministic system. In such a case, the utilization is 100% since transfer of work from one station to the next proceeds in a synchronized fashion. (ii) Imbalances work to strongly reduce the throughput. Such imbalances can arise from stochastic breakdown of machines, differences in processing times of individual units, and inadequate coupling (or feed through) between stations. (iii) In a real world situation, all manufacturing systems are stochastic due to random and unpredicted machine breakdowns and faults. Furthermore, the units used in any real processing line are dissimilar and do not have identical characteristics. For example, the down times of the various machines is very likely to be different, with some units which perform complex tasks likely to break down more often than others. Similarly, the conveyor lengths in any actual manufacturing plant are not likely to be identical based on layout and space considerations. Thus, non-uniformity is the norm. (iv) Consequently, the simple formulas and closed form analytical expressions given in the previous section is not really valid for practical systems and cannot be applied. For example, the $N-1$ identical buffer sections often assumed in previous formulations do not hold.

In view of the above, it becomes obvious that instead of approximate closed-form solutions, more rigorous numerical techniques need to be used for accurate modeling. For this reason, a numerical simulation scheme is used here in this thesis work. The numerical simulations have to include the stochastic aspects of machine breakdown and failure, their down times and probabilities of defective parts. In addition, the actual system with its variations in buffer capacities needs to be included. Finally, the simulation model has to adequately represent a tightly-coupled system. Only by so doing would it become possible to include the starving and blockage aspects into the process flow. Simulation details are given in the subsequent chapter.

CHAPTER III

DETAILS OF THE SIMULATION TOOLS USED IN MODELING

3.1 INTRODUCTION

Simulation is commonly defined in the dictionary as “an attempt to predict aspects of the behavior of some system by creating an approximate model.” This can be done through physical modeling, by writing a special-purpose computer program, or even using a more general simulation package that is geared towards a particular kind of simulation such as structural engineering, fluid flow, etc. Typical examples include the use of aircraft flight simulators or electronic circuit simulators. Many of the real time simulation applications have recently begun to rely extensively on computers as a consequence of the increased computing power of these machines that has come about through advances in technology. The discipline of routing has made active use of this development and presently harnesses animation to improve the simulation style and capability. Many routing packages have been developed to analyze the data, construct routes and to obtain the optimal solutions for the routing problem. The routing practitioner has a choice to choose among the numerous packages available to him depending on the characteristics of each package and the complexity of the problem at hand. Some of the popular routing packages are “Georoute” from Kositzky & Associates Inc. [29], “Load Manager” from Roadnet Technologies, Inc. [29], “RIMMS” from Lightstone Group, Inc. [29], RoutePro from Caps Logistics, Inc. [29], and Routronics from Carrier Logistics, Inc. [29]. Although most of these packages share some common basic characteristics of a vehicle routing problem, they also have some unique features. With the availability of a

multitude of simulation packages, it becomes important to judiciously select a package based on its features, the compatibility with the problem at hand, and the solution capabilities offered.

There are also many commercial software packages available for the process flow problem. Therefore, careful research and problem evaluation has to be carried out *a priori* before deciding on and selecting the package. A few relevant issues that need to be taken into consideration in the selection process are listed below.

- *Problem identification and definition:* Identifying the problem is considered to be the first step for any model can be constructed or a simulation performed. It is important to understand the definition of the problem, and also to identify the complexity of the problem before implementing all data into the simulation model. Understanding the problem helps in selecting the proper tool for the simulation.
- *Understanding the system and determining the goals and objectives:* Understanding the system is the second step that is involved in the model formulation. In this present research, a thorough understanding of the manufacturing plant and the routing problem is necessary. Designing a mathematical model and proceeding with its numerical implementation without understanding the system and its underlying characteristics can lead to a serious mistake [30]. Once the problem has been defined and an overview of the system to be modeled is known, the goals and objectives for the problem have to be

determined. The model should be formulated and the simulation should proceed in keeping with the project goals and objectives.

- *Selection of the right package:* A study into the various simulation packages has to be done before selecting any particular package for simulation purposes. This will ensure that the right simulation tool for a given problem and objective has been selected. In addition to the above criteria, the tool should be inexpensive, be easily available, and allow for an easy construction of the model.

Once the package is selected and a model is constructed, all relevant data then has to be obtained. This can be a slow and tedious gathering process. The data collected is then used as an input to the simulation model and corresponding results obtained for analysis. This result should match the stated project goals and objectives. Usually, an initial base model is constructed to verify the effectiveness of the software. If the model output matches the expected data and exhibits the expected trends, the overall process is deemed successful [31]. Otherwise, the model is modified to be consistent with the desired output.

3.2 FEATURES OF VARIOUS ROUTING PACKAGES

Simulation is a technology that will continue to evolve as related technologies improve and more time is devoted to the development of the software. Despite this dynamic evolution, some general desirable features of any simulation package always remain. These, in general, can be summarized as follows [32].

- Input data analysis for distribution fitting for discrete event simulation.

- Components and templates reusability.
- Two and three-dimensional animation.
- Online help and tutorials for the package.
- Interactive debugging.
- Automatic model generation.
- Output analysis tools.
- Optimization.

In addition, it is very likely that any given simulation package will have its own unique characteristics. Some such specific features [33] for a routing package are listed below.

Good Graphical User Interface (GUI): An environment portraying routing operations and allied details such as sites, orders, products, vehicles, etc., in a graphical user-friendly context forms part of the major requirement in the routing package. Interfaces such as pop-up menus and combination boxes in conjunction with the above make the front end visually animated and palpable to the user's imagination.

Database Connectivity: Organizations have enormous amounts of data pertaining to locations, customer information, depot details and employee details. It is not practical to collect the same from discrete parts of the system and insert them into one database. Thus, an efficient database link between the various discrete parts of the system is required failing which the package becomes void or inefficient.

Efficient Algorithms: Heuristic methods and algorithms are at the heart of any routing package. The routing package is useless without the algorithms. A case of routing

different scenarios should be handled. Multiple algorithmic methods to solve a variety of routing problems are needed. Computer-processing time is an important factor in routing and the algorithms used should be efficient.

Demographic & Geographic Catalog: The term “routing” defines the dispatch of vehicles through proper cost-minded routes. Thus, any routing requires the geography of the particular area or region. Consequently, the routing executrix should have all the available maps that portray in the finest detail possible, the roads, all connecting transit and rail links, etc. Hence, the involvement of a geographic data catalog is a necessary and primary requirement for the routing package.

Applicability: The ability to handle a versatile routing environment such as dedicated deliveries, dedicated pickups, combinations of the above, single or multiple depots, backhaul routes, frequent-stop routes etc. is necessary.

3.3 ADVANTAGES AND BENEFITS OF SIMULATION SOFTWARE

Use of simulation tools helps in avoiding the expensive, time consuming and disruptive nature of the traditional trial-and-error techniques. Some of the advantages of simulation and numerical modeling are given below [32].

- The greatest advantage is that simulation allows the scientist to model complex and dynamic phenomena that otherwise could not be dealt with any other way.
- By simulating the system, the scientist gains valuable insight into the system and into the relative importance of the different variables.

- Simulation allows for the compression of real time. To predict the behavior of a system over the period of a year may take only a few minutes using computer simulation.
- To comprehend the basic concept of simulation does not require a very deep knowledge of mathematics. It can be used as a decision-making tool.
- Using simulations to model a system before it is built avoids many pitfalls that may otherwise have been encountered during actual start-up.

The general characteristics of a simulation package that make it a very powerful and useful tool in problem solving can include the following:

- 1) Interdependencies of the system can be encapsulated.
- 2) The model is versatile to any system.
- 3) The system takes into account the variety in the model.
- 4) The package is inexpensive, less disruptive, requires less memory and runs faster than testing on the actual system.
- 5) The package clearly depicts the characteristics of the system over time.
- 6) The package provides information on multiple performance measures.
- 7) Animation is realistic and pleasing.
- 8) The package provides convincing results that are accurate and is easy to communicate and understand.
- 9) The package is capable of focusing any particular area of interest.

Since simulation accounts for interdependencies and variations, it provides insights into the complex dynamics of a system that cannot be obtained using other analytical

techniques. Simulation promotes an easy “try-and-see” attitude, stimulates innovation and also takes out the emotion of the decision-making process thereby providing realistic solution to any problem.

3.4 THE ROUTE-PRO SIMULATION MODEL

RoutePro is a powerful routing software technology incorporating all the features of a good routing package. RoutePro is composed of two routing applications. They are RoutePro Designer and RoutePro Dispatcher. In this study, the RoutePro Designer has been used.

RoutePro Designer: This is the strategic tool used to optimize service strategies, balance workload, and design master and zone routes. This is used for the following purposes:

- 1) To determine when sites should be serviced in order to balance the workload performed by vehicles.
- 2) To design master routes that captures the usual stops at customer and supply sites (based on historical or forecasted workload).
- 3) For building optimal routes that service certain geographical areas and work zone.

RoutePro Dispatcher: This operational tool is used to schedule vehicles and drivers on optimal multi-stop routes and to track their progress as they complete the route. This is used in situations for purposes such as:

- 1) To incorporate daily order variations into master routes.
- 2) For building optimal routes that service specific customer locations.

- 3) To export route data to external applications, e.g., billing, auditing inventory control system.

The basic features and capabilities of RoutePro are as given below.

- (i) Optimization Technologies: RoutePro has the backing of the best algorithmic and heuristic techniques to solve most routing problems. It has a set of eight routing methods, which employ the best routing algorithms.
- (ii) Map based graphical user interface: RoutePro allows one to visually represent the routing operations in a map-based context. It has pop-up menus, allowing one to access the menu commands that are pertinent to the selected object. The graphics make it easy to visually analyze the configuration and performance of the routes.
- (iii) Time-based Schedule Windows: RoutePro has a provision to help the user to visually examine routes along the dimension of time. The Schedule Windows functionality inherent in RoutePro can display routes as Gantt charts, complete with projected start and end times for each activity, and a graphical indication of whether the activity falls within the service time window. The Asset Scheduling Window is another functionality that displays routes, drivers and vehicles in a time-based context. From this schedule window, one can “drag-and-drop” drivers and vehicles onto routes.
- (iv) Data Sharing: RoutePro has the support of TOOLKIT database connection technology, which helps one to share business data with other applications. For instance, one can use the industry standard query language, SQL, to retrieve data from an external data source as well as update it by our solution.

(v) Geocoding Technology: RoutePro provides a variety of geocoders like postal and Etak geocoders. The geocoders support both batch and interactive geocoding. There is also the provision for reverse geocoding, thereby updating the address in a site file with information from a reverse geocoder database.

(vi) Online Geographic Catalog: There is a large collection of maps, road files and street networks provided by CAPS Logistics, which are quite necessary for routing.

(vii) Report generation: With RoutePro, one can generate a variety of professional style reports. These reports are in Microsoft Excel, which is the standard today. One can also create business graphics, such as pie charts and bar charts, to visually analyze the data and solutions.

(viii) Applicability: RoutePro can handle any operational environment, which includes dedicate deliveries, dedicated pickups mixture of pickups and deliveries, etc. This makes it quite versatile.

(ix) User Assistance: RoutePro has an extensive help file, which provides both context-sensitive and task-oriented help. With the context-sensitive help one can access information on a particular element in a dialog box, and with the task-oriented help one can have step-by-step instructions. There are also electronic guides called wizards, which lead us through the process of setting up a routing system, creating data files and routing. Lastly, there is electronic documentation in an electronic book format.

An integral part of any routing system is the presence of Logistics objects [54]. RoutePro incorporates many such objects. Each object is a type of file that one loads or creates in RoutePro. The objects are as follows:

- (i) Sites: All the information regarding the sites in a routing operation is contained in the site file. Depots, distribution points, customers, etc., are a few of the entities present in the site file. It generally contains the site type, address, its latitudinal and longitudinal location, and contact information of that site. This is part of the basis of the routing model.
- (ii) Orders: The orders from the various customers are contained in this file. It has information on the deliveries, returns, pickups, and backhauls, if applicable. The order file shows the amount of products a customer needs and the location of the pickup and delivery points.
- (iii) Order Line Items: The details associated with each order are contained in the order line item file. One order may contain many distinct items. This file gives us a detailed picture of these orders. This file is optional for a routing operation, though it is preferred to have a detailed perspective of the orders.
- (iv) Time Windows: Time windows are valuable for a routing operation. The period of time during which a pickup or delivery should occur is defined as a time window. The dates and times of the pickups and the deliveries are contained in this file.
- (v) Products: The product name and unit of measure is contained in the data regarding all the products in the routing operation.
- (vi) Vehicles: The vehicular technicalities encapsulated in the file type involve details such as the number of vehicles per fleet, the type of the vehicle, the size of the vehicle, and the utility of the vehicle. Some pieces of data that are in the file are fixed cost, cost per distance and maximum unit capacities. This is a required file in the routing model.

(vii) Drivers: This file contains the list of all the drivers who are available to run the routes. The data that are in this file include driver's name, permit type, seniority level, wages, work hours, and base location. Drivers can be associated with the routes after the routes have been created. This is not a required file since a default driver can be mentioned. The default driver represents all the drivers.

(viii) Routes: These files are automatically created when the routes are created after using an algorithm. Data in this file includes the start location, drive time between stops, estimated time of arrival (ETA), estimated time of departure (ETD), and end location.

(ix) Maps: This may not be a required field for routing operations. However, to give the routing a geographic view and to make it more presentable, map files should be loaded in the project. There are two different types of maps. They are Raster maps and Vector Maps. Raster maps are scanned images of paper maps. Vector maps are considered smart maps one can view the data that makes the maps. In the vector maps, there are background maps and detailed maps. The background maps consists of two types of files, the map file, which makes up the boundary of the map, and the map zone file, which makes up the fill within the boundary. The detailed maps are available from well-known commercial vendors such as ETAK and MapInfo.

(x) Road Networks: In order to simulate a real road network, the road network file has to be loaded into the routing model. The road network consists of two files, the road node file which contains the intersections that link the roads and the road link file which contains the actual roads. Another file, the access road file, is needed in order to connect the sites to the closest road nodes in the road network. These access roads allow a vehicle

to leave the road network to visit a stop on a route. The access road file is generated automatically and is invisible to the user.

(xi) Rush Hour Windows: This file contains the dates and times when the road network speed should decrease due to rush hour. The data includes the dates and times of the rush hours and the factor by which the speed should decrease.

3.5 ROUTING ALGORITHMS

There are different algorithms that the RoutePro software uses to give an optimized solution to a given routing problem. The algorithms used are the backbone for any routing software and so the algorithms play a very important role in determining the efficiency of the software. The algorithms should take into account all the constraints that the user gives and should be able to produce the best-optimized solution. RoutePro is an efficient tool as the package has eight different algorithms to solve a particular problem. A brief description about each algorithm is presented below.

(i) Boundary algorithm: This algorithm is very efficient for the orders that originate from or destined to a single depot. The orders can be dedicated pickups, dedicated deliveries, or a mixture of pickups and deliveries. The basic idea about this algorithm is that it builds the routes simultaneously [35], starting with highest priority order. Then, the algorithm proceeds with the next priority order and continues to find the best route. If the best route already exists, the algorithm starts looking out for the best possible location. There are different options available to give priority and the algorithm follows the priority accordingly. The priorities include, 'Order priority', 'Furthest distance', 'Most cubic feet', 'Shortest pick-up time window', 'Shortest delivery time window', 'Earliest pick-up

time window', 'Earliest delivery time window'. Based on these priorities the algorithm builds the route. The unique feature of the boundary algorithm is the ability to construct the routes simultaneously. Let us suppose that the orders be prioritized by further distance, the algorithm starts building the route with the farthest distance first and creates a route with that order. Then, the algorithm picks up an order of the next highest priority and starts a route with that order. But if the order can be fixed in an already existing route, then it is inserted at the most appropriate place in that route. So all the routes are formed first and then the orders are added to the routes. There are two important parameters that determine the formation of routes with this method. These are given and discussed below.

(a) Initial and Maximum routing angles: The minimum angle within which all the orders are routed is the initial angle and the maximum angle is the maximum angle within which all the orders are routed. The angles mentioned here are obtained by drawing a horizontal straight line from the base site or the seed site. An arc is drawn from here with a particular radius and this can be drawn both in clockwise and in anti-clockwise directions. Any order not falling within this sweep is added to the route provided that the time mentioned in the time window constraints is met. The idea of the angles was based on the theory of the sweep algorithm. In the sweep algorithm [36], the vertices or the sites are represented by their polar coordinates (θ_i, ρ_i) , where θ_i is the angle and ρ_i is the ray length. An arbitrary vertex is taken and assigned the value zero for its angle, and from it the angles of the other vertices are determined. In the case of the boundary algorithm, the seed site is assigned the angle zero and based on the seed site; the angles of the other sites are determined.

(b) Radius in miles around the depot where the angles are not enforced: The area drawn around the depot where the angle constraint is not enforced. This value should not be kept very high because there is a high probability that all the orders come under this area thus the boundary algorithm becomes void. Also, this is not practically feasible.

Therefore, proper selection and adjustment of these parameters has to be carried out in order to get the best solution from this algorithm. Once these parameters are chosen and the routes are stated, an insertion method is used to insert the best orders in the routes. By doing so if the orders fail to meet the other constraints like time, etc., then that particular order is not routed.

(ii) Nearest Neighbor Algorithm: Unlike the boundary algorithm, this algorithm builds the route one at a time. Here the vehicles that are used for routing are filled to its maximum capacity and so one vehicle is used first and once this vehicle has finished routing, the second vehicle is chosen. Thus, this algorithm builds one route at one instance. Building of routes will start from the order having the highest priority. Then, the algorithm builds the route by inserting the unrouted order and nearest to the current order site. Similar to the boundary algorithm, nearest neighbor algorithm [37] also has two important parameters as discussed below.

(a) Initial and Maximum routing angles: These parameters are similar to the parameters used in the boundary algorithm. The orders should fall within the given angles here for it to be routed. If not, these orders will remain unrouted.

(b) Distance from the seed site: This is the maximum distance of the site from the seed site. If an order is farther than the distance specified, then it will not be routed. The important parameter to be chosen here is the seed site. The site, which has the highest priority order, is selected as the seed site and the distance to the other order sites are measured from here.

Once the route is complete, another seed site based on the priority is formed and the routing is started. The basis of this algorithm lies in the Dead Sweep Heuristic (DSH) [38]. The details about this algorithm are described in the second chapter. In the DSH orders are added to the route based on a cost function, which takes into account the distance, urgency and the angle of that order with respect to the last customer visited. Weights are given to each of these parameters based on their significance. The Nearest Neighbor algorithm works in a somewhat similar way. However, the nearest neighbor algorithm, the angle and the distance are taken into consideration first. The angle in the nearest neighbor algorithm is also not the angle with respect to the last customer, but it is with respect to the seed site. This is fixed. So after these two criteria are met, it starts building the route. When it is in the process of adding a customer to the route, it checks the site's time windows. If the time windows suit the route's time, then the order is added to the route. Another important parameter is the amount of time that the vehicle can be on road, which is also specified as a constraint. The algorithm can be implemented in the following steps.

Step 1: Begin the route starting from the depot.

Step 2: Look at the priority list and select a seed site based on the highest priority order.

Step 3: The sites nearer to the seed sites are routed and also satisfying the conditions given in the initial and maximum angle and also maximum distance from the seed site. If the conditions are met the site is added or the site is left unrouted.

Step 4: Start a new route by selecting another seed site.

The nearest neighbor algorithm has two important parameters, distance from the seed site and the angle from the depot. By varying these parameters, one can test their significance in the algorithm.

(iii) **Cluster Algorithm:** The Cluster Algorithm [39] tries to build one route at a time similar to the Nearest Neighbor algorithm. The algorithm initially groups the sites into clusters and then inserts the sites that are near to this cluster into the routing. Visiting the sites that are close to each other carries out this routing. The algorithm will then grow by inserting all the unrouted nearby sites closest to the cluster. This algorithm is used when the orders must originate from or be destined to a single depot. Some of the parameters that are involved in the cluster algorithm are the Tree, Spanning tree, and the Minimum Spanning tree (MST).

(a) Tree: The tree is a connected graph. The tree has one and only one path joining any two of its vertices. A graph is a tree if it has N vertices and $N-1$ edges.

(b) Spanning tree: A sub-graph which is tree and which spans all the vertices of the original graph is called a spanning tree.

(c) Minimum Spanning Tree: The tree with the least total weight among all the spanning trees of weighted and connected graph is called Minimum Spanning Tree.

The procedure involved in this algorithm is given below.

Clustering: Initially a graph is drawn with the sites as the nodes and the edges being the distances between them. The edge weights are calculated using a shortest path algorithm on a hierarchical street network. Then the objective is to find the cluster of sites involved in the routing. The structure of the Minimum Spanning Tree (MST) is employed to find regions where the sites are positioned together. The idea is to divide the MST into sub-trees, where the nodes of each sub-tree represent the customers belonging to one tour.

Greedy Algorithm: A greedy algorithm [40] is employed to calculate a partition of tree into valid sub-trees. The MST is calculated by Prim's algorithm [41]. This algorithm generates the tree in the required structure, i.e., a rooted tree where the tree is chosen by the algorithm.

The sum of the weights in a sub-tree is defined as follows

$$W_i(v) = \sum_{v \in V(T(v))} w_i(v) \quad (3.1)$$

where $T(v)$ denotes the sub-tree of T rooted at the node v .

In a bottom-up scanning algorithm, the branch with the highest weight $W_i(v)$ is cut off recursively. Fig.3.1 illustrates the recursive cutting off for one weight function. For each node, the numbers in the circle represent the weight values $w_i(v)$, whereas outside the circles the values $W_i(v)$ are denoted. The restriction is given by the upper-bound $R_1 = 15$. The weights are summed bottom-up from the leaves to the root. If $W_i(v) > R_1$, the successors v_i of v with the highest $W_i(v_i)$ -values are cut off until $W_i(v) < R_1$ for the remaining sub-tree $T(v)$. The idea is to scan the tree bottom-up and to cut off the branches if the bound is exceeded. This is the basic procedure used to build a cluster.

Routing: For each cluster thus obtained, new sites are added depending on the distance it is from the cluster. Another important parameter that should be taken into account is the time constraints that have been given as a data. So the routing should also take this into consideration and a perfect routing following all the time constraints as well as inserting as many sites as possible into the cluster. Thus, an effective routing is carried out using the cluster algorithm.

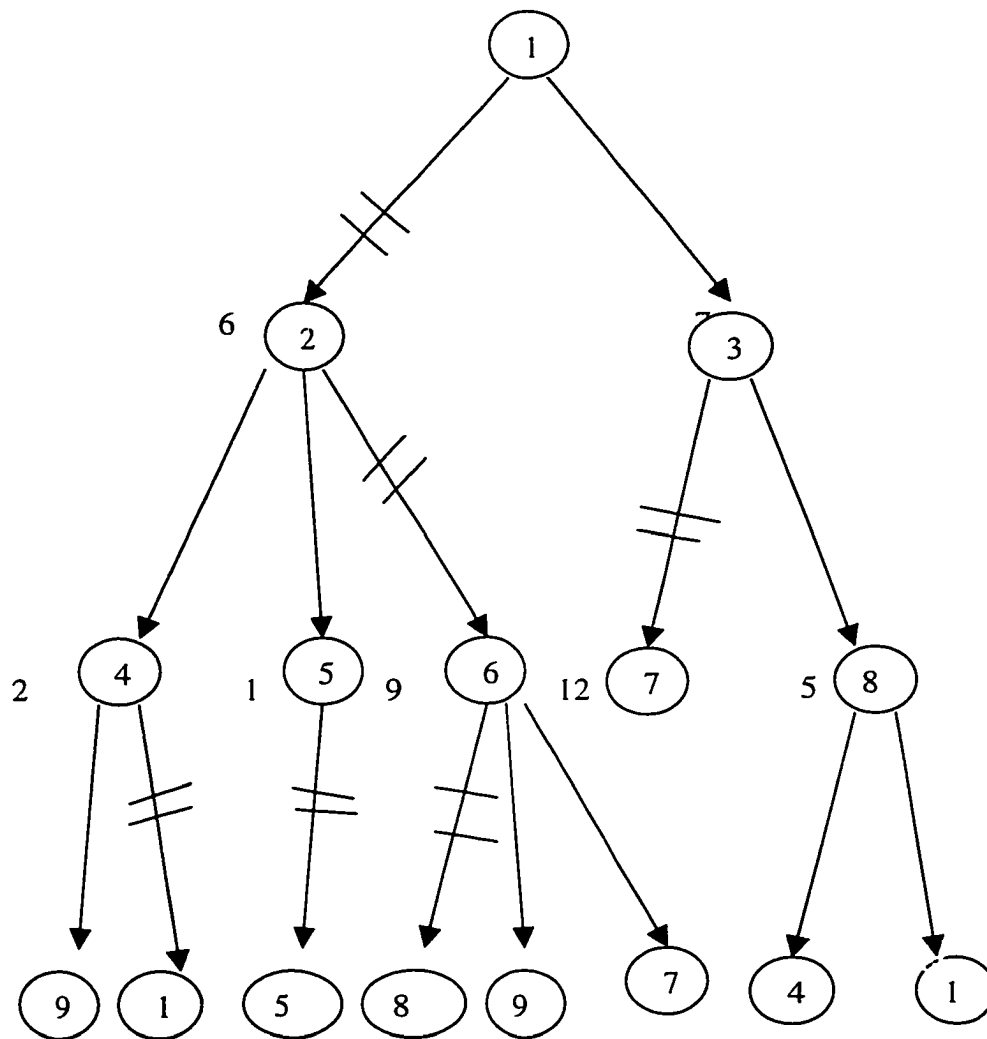


Fig 3.1: Greedy Algorithm for one weight function.

(iv) **Nearest Pass Algorithm:** The Nearest Pass Algorithm [42] builds routes one at a time and tries to build a full truckload with one vehicle before using another vehicle similar to the previous mentioned algorithm. The highest priority order will start a route. The algorithm will then grow the route by inserting the unrouted order that has the least route cost.

The major difference between this algorithm and the Nearest Neighbor algorithm is that this algorithm is used when one has orders that originate from or are destined to multiple depots. The orders can be dedicated pickups, dedicated deliveries, a mixture of pickups and deliveries, or an interspersed of pickups and deliveries.

(v) **Pie Algorithm:** The Pie algorithm [42] builds routes simultaneously. The highest priority order will start a route. The algorithm then looks at the next highest priority order and finds the best route for this order. The best route for this order may already exist in which case it is inserted at the best possible location on the route. If the best route does not already exist for this order, a new route will be created. The algorithm goes through the order priority list and finds the best route for each order.

This algorithm works well for routes originating from or destined to multiple depots. The orders can be dedicated pickups, dedicated deliveries, or a mixture of them. This is very similar to the boundary algorithm in theory except that this algorithm handles multiple depot routes.

(vi) **Profitable Backhauler Algorithm:** The profitable backhauler algorithm [42] builds the route on the existing routes only. The algorithm only looks at orders that are marked “For

Profit.” These orders should have a revenue value. The algorithm will add the backhauls to existing routes if the revenue is greater than the cost.

This algorithm works well when there are orders originating from or destined to multiple depots. There is a separate column for revenue in the orders table. If a particular order is marked as a revenue generating order, it is eligible for routing with this algorithm. This is useful only when there are revenue-generating backhauls and is possible only when the company uses its own transportation.

(vii) Tree Algorithm: The Tree algorithm [42] builds routes one at a time and tries to build a full truckload with one vehicle before using another vehicle similar to the cluster algorithm. The highest priority order will start a route. The site that starts the route is called the seed location. The algorithm grows the route by adding the unrouted order whose summed distance between the location of the site order and the seed location and the distance between the depot and the site order is the smallest.

This algorithm works well for orders originating from or destined to a single depot. This algorithm is useful when there is a need to minimize empty travel on the first and/or last legs of the route.

(viii) Weighted Constraint Algorithm: This algorithm [42] builds routes simultaneously similar to the boundary algorithm. The highest priority order will start a route. The algorithm will then look at the next highest priority order and finds the best route for this order. This algorithm has constraints such as vehicle capacity and time windows, to name a few. This algorithm is for orders originating from or destined to multiple depots. This

algorithm is suited for filling existing routes. Sites are routed initially using any other algorithm and then this algorithm is used to fill any unrouted orders from the previous algorithm.

These are the various algorithms used in this package and make this a powerful tool for any vehicle routing problem. It is advisable to obtain the solution from all the algorithms and take the solution that is optimized to one's needs.

3.6 REVIEW OF ARENA

This section explains in detail the advantages and the features present in Arena that call for its selection as the ideal software for solving discrete-event problems in manufacturing and process flow. Arena simulation software creates animated computer models that accurately represent virtually any manufacturing system. First released in 1993, Arena is a flexible and powerful tool. It has an object-oriented design and the unique ability to be tailored to any application area. Arena also is very easy to use; with its point and click interface and fill in the blank dialogue boxes, there is never a need for programming. The important feature of Arena is that it allows for discrete inputs and takes into account random, stochastic breakdown and failure events in a sequential process flow environment. The simulation software has the features of graphical model construction, input data distribution fitting, output analysis support and animation of the output. Also, Arena can be used for either discrete event simulation or continuous event simulation. This project involves discrete event simulation as discrete parts are being produced and sent to the next processing machine or section based on uniform process

cycle times. Animation is important feature of the software and it allows the user to develop complex, complicated well-animated models using this software. A brief overview of the software is given below.

3.6.1 Configuration of Menus

Arena software looks very similar to any other application running on the Windows platform. The menus that are available of opening the Arena applications are: File, View, Tools and Help. Once a file is opened or a new file is created, few other menus such as Edit, Arrange, Module, Run and Windows menus are added. A concise review of the feature present in each menu is presented below.

(i) File menu: The file menu is the place where an Arena model is either created or opened. The file menu also helps in importing any drawing from AutoCAD or any other CAD program software. Apart from these functions, the file menu has the option of saving the opened file, printing the model, and remembering a few of the recently opened files. A special command in the file menu is the “send” command. The send command allows sending mail from Arena and attaches the created model with an appropriate message. Finally, the exit command quits out of the model.

(ii) Edit Menu: Here all the usual functions present in most of the edit menu such as cut, copy, paste, select all, deselect all, redo and undo. Apart from these standard commands, other special commands like object properties, such as giving unique tag for each object are also present. If the present model has link to any other files, such as spreadsheet or

sound file, the command “Link” helps in identifying them and if required modify them. Also, a new object could be inserted into the present module using the Insert command.

(iii) View menu: The view menu is used to control the view of the model window. Any adjustment such as adjusting the altitude of the model can be done using the “Zoom” command. Zoom command has a zooming factor, which is controlled to resize the module. There is a sub command View inside the view command offers certain “canned” views of the model. There is also a “Named” view, which allows one to define, change and use one’s own views. Grid and Snap are useful if one wants to line things up geographically [43]. The view menu has a Toolbar command by which one can designate which sets of buttons are displayed on the screen. The status bar gives the coordinates of the mouse on the model.

(iv) Tools menu: Tool menu gives additional capabilities to the software. The Model-Jump-start wizard constructs a model based on the input given to wizard. This gives an idea about the model that can be constructed for different inputs. The next command is the Input Analyzer, which gives a distribution fit for the input data. This distribution is given as the input in the model that is created. Thus, the stochastic nature is introduced to the model. There is also an output analyzer that gives the result of the model in a graphical way. The required result and also the desired type of waveform can be selected so that the result can add some additional meaning. The next command is the Scenario Manager that lets us run a batch simulation. There is another command, “Show Visual Basic Edition,” where in a visual basic program can be written to accompany the model.

Finally, the Options command lets one to customize a lot of how Arena works and looks to suit the need of the model.

(v) Arrange menu: The Arrange module pertains to the position of the module and the graphics that are involved in the model. The model can be brought front or back using the commands in the Arrange menu. It is used to group, ungroup and rotate the image in any given direction. The “Grid” command forces the selection to align to the underlying grid of points.

(vi) Module menu: The basic operation performed in this menu is to relate the logic used in the module to templates associated with the various modules. The Connect command sets up a graphical connection between the modules for entities to follow. Auto-Connect is a toggle that allows an automatic connection between newly placed modules and those already selected. Smart Connections caused newly added connections to be drawn in three lines instead of one. Template panels attaches or detaches any template required for the model.

(vii) Run menu: This menu is used for the execution of the simulation model. There are a few options such as “fast forward,” “check model,” “speed setup,” etc., that add features to the run command. There are also few other commands such as Break, Trace and Watch, which are used to run the simulation until a certain stage. These functions are useful while debugging a program with many lines.

(viii) **Window menu**: This menu is used when one has to open two or more window at the same time in the same model. The windows are arranged physically in an overlapping Cascade, or in a non-overlapping tile arrangement. The Arrange command organizes each window when several windows are minimized. Also, the physical appearance of the window such as the background color can be selected using the Use System Background Color option.

(ix) **Help menu**: The help menu is one of the several online resources that can be accessed from the Arena software. The help menu contains a help topic that consists of a table of contents about the software. There is also a Find utility for knowing about the topic we want. Other help features that add color to the help menu are the Introduction, Tip of the Day and About Arena, which opens with a welcome window and from which various help and information can be obtained.

3.6.2 Configuration of Toolbars and Templates

Arena utilizes a number of toolbars and templates. These are discussed next. Toolbars are group are buttons or drop-down menus that helps in quick access to common activities. There are several toolbars present in the View menu. Some of these are listed next.

(i) **Standard toolbar** – does the operation present in the File menu such as open of an existing model, new model, etc.

(ii) **View toolbar** – to zoom in and out particular portion in the model and also to select and deselect some parts in the model.

- (iii) Arrange toolbar – to bring selected objects to the front or to send them back, group or ungroup a set of images.
- (iv) Draw toolbar – contains various standard figures and shapes and also contains buttons to draw a line, curve, etc.
- (v) Color toolbar – contains a set of color and these colors can be chosen to change the color of the background and foreground.
- (vi) Animate toolbar – contains capabilities to allow animation to the model or enhance the animation that is inherent in some Arena modules.
- (vii) Template toolbar – contains the modeling constructs for Arena, organized into panels. These panels contains the templates such as Common, Support, etc.
- (viii) Run toolbar – this toolbar contains buttons to start the execution of the simulation, Fast-forward, Go step, Pause and Stop operations.
- (ix) Run interaction toolbar – has buttons for checking the model, Break, Trace and Watch from the Run menu.
- (x) Integration toolbar – contains buttons related to Arena's Module Data Transfer wizard and VBA (Visual Basic Editor).

Among these toolbars, the template toolbar is significantly used and acts as an important building block for any model. The various templates used in Arena software are: Block, Common, Elements, Support, Transfer and UltArena. The functions of the various templates and the object present in each template are described below.

(i) Common template: The basic blocks required to model any manufacturing plant are, the Arrive block, the Depart block and the Processing block. These three forms the basic block while other complex designs are built from these blocks.

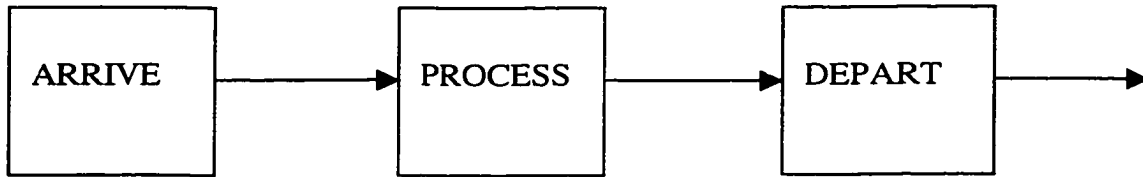


Fig 3.2: Basic building block of any model.

The raw materials arrive at the Arrive station and are passed on to the Process station and the processed material comes out at the output at the Depart station. The common template contains modules that perform these basic operations. Each module such as Arrive, Depart, etc., has a set of sections that are to be entered. These sections may be the name of the present station, name of the next station, process time, time to reach the next station, etc. Distribution functions can be given to the processing time and also to the uptimes and the downtimes of the machines, thus introducing stochastic behavior to the data. The uptimes and the downtimes are present in the machine, which is indicated as server in the software. All these components have queue associated with them and it represent the number of parts waiting on the queue to get processed.

(ii) Block template: The components contained in this template are useful for low-level construction of the model. The various components contained in the Block template are Access, Allocate, Branch, Copy, Group, Queue, Request and Scan modules. Each of

these modules performs unique functions and is used according to the necessity of the model.

(iii) Transfer template: The basic components present in this template do the operation of transportation of the material from one station to another. Although all the components perform similar operations, each module has a special characteristic that are different from the rest of the module; so, care should be taken to chose the right module for the specific application or function. The various modules present in this template are Convey, Access, Allocate, Activate, Conveyor, Distance, Exit, Segment, Transport, Route, Free and Halt.

(iv) Support template: This template has modules that performs operations similar to those present in the common panel but has lower performance utilities. This is used when the model does not require high performance modules; these modules combine themselves and work to function as the big module. The components that are present in the support module are Batch, Choose, Count, Delay, Read, Dispose, Duplicate, Release, Seize and Station.

(v) Element template: The components in the element template are less useful than in the other templates. It contains modules such as Tallies, Dstats (Discrete-Change statistics), Frequencies, Pictures and Queues.

All of the above-mentioned templates are useful in the construction of the model for any manufacturing plant. The simulations are typically run for long periods of time until a steady state condition is obtained. The results obtained are available either numerically in terms of raw data or graphically through plots and charts.

CHAPTER IV

RESULTS AND DISCUSSION

4.1 INTRODUCTION

This thesis research, as detailed in a previous chapter, has focused on two separate, yet similar, simulation problems. The first task was to analyze a routing problem and to work to obtain an optimized solution. The second dealt with the simulations of a production process for increasing the throughput by incorporating suitable buffers. The primary objective of the first project was to explore various delivery routing strategies and obtain routes that would minimize the total network mileage, while maintaining the specified “pick-up” time slots requested by the various clients. A related task was to evaluate the possible advantages of centralizing all routing activity from a single site at the DePaul hospital, instead of the two-hub scenario (DePaul and Maryview Hospitals) currently in effect. A total of eight different scenarios were studied. Two were for the current delivery routes and were designed as a baseline check. Two other simulations were carried out to optimize routing for the current two-hub situation. Four more simulations were performed for routing evaluations of a centralized DePaul hub for a variety of constraints. In two of them, the total “time-on-the road” for any vehicle was set at a maximum of three and four hours, respectively. This is roughly in keeping with the temporal spread in durations for the current trips. In another scenario, some slight variations in the “pick-up time slots” were allowed for a three-hour run. Finally, an optimized four-hour run with significant changes to the “pick-up and delivery schedule” was evaluated. One of the main advantages of the software Routepro is the incorporation of various algorithms that are used for routing and optimization. Each of the eight

algorithms was used for the simulation and the algorithm that gives the best solution was selected.

The other project involved optimization of process flow in a manufacturing plant using the Arena software tool. Siemens Automotive, Inc., has been looking for ways to improve its in-house processing throughput and overall efficiency for enhanced profitability and growth. A specific goal is to focus on a production enhancement scheme that incorporates a “buffer” into the existing assembly line. This should allow for a more flexible supply and overcome the problems of “starved” or “blocked” assembly line sections. Given the current space constraints and machine throughput data for the DEKA manufacturing system, there appears to be one possible location for such a buffer. However, the relative benefits of placing a buffer at this location were not known, nor had there been a determination of its optimal capacity. In this project, discrete event simulation studies based on the ARENA software tool were carried out.

4.2 PROBLEM #1: INTRODUCTION AND BACKGROUND

Bon Secours uses a heterogeneous fleet of couriers and self-owned vehicles for pick-up and delivery of medical specimens, supplies and related paperwork to a wide network of doctors in the Hampton Roads area. The current practice involves delivery and pick-up from both the Maryview and DePaul hospitals based on a two-hub system. There are two primary constraints for this pick-up and delivery problem: (a) The vehicles should return within a reasonable (three-four hour) window so that laboratory processing on the samples collected can occur in a timely fashion. Delays will not only degrade the specimens, but also create backlogs at the laboratory. This would then preclude timely

analysis and the transport of reports back to the doctors' offices for quick resolution of the patients' cases. (b) The routing process must meet the set "time-windows" for pick up and delivery at the various doctors' offices. These times reflect their preferences in accordance with their schedules. There is a general requirement to get the delivery to all the locations in keeping with the constraints with minimal travel mileage. Thus, suitable routes have to be constructed which then will reduce the mileage costs, wear and depreciation on the vehicles, and improve profitability of the organization. Enhancements in efficiency and greater manpower/fleet utilization should also result. Of special interest is the consolidation of the DePaul and Maryview hubs into a single centralized DePaul site. Thus, the operational and logistics issues that the management is interested in examining include:

- (1) Possibility of reducing the number of routes and/or total mileage for cost reductions.
- (2) Quantitatively evaluate a rearrangement in which all routes start and end at a single DePaul site.
- (3) Evaluate the relative trade-offs between three-hour and four-hour runs.
- (4) Quantitative comparisons of a few selective routing options.

At the start of the project, the potential benefits of a routing reorganization were not known, and an evaluation was necessary before investing into routing changes. Several hypothetical scenarios needed to be evaluated quantitatively in terms of reductions in mileage and labor costs. A study was carried out to assess a number of scenarios based on

numerical simulations. In the process, an optimal routing strategy can be generated for delivery and pick-up prior to costly implementation.

4.2.1 DATA SHEETS AND DIALOG BOXES IN ROUTEPRO

Before getting on with the discussion of the project, a brief summary on the data entry boxes available in Routepro software is given.

Data Entry Boxes: The most important and often used dialog box is the System Properties box. This data box controls all the key factors of this software. The System Properties box consists of six other sub data entry boxes. Of these, Data and Routing are the most frequently used. Data allows the user to access site types, order types, fleet, quantities, etc. It is necessary to generate the fleet every time before any simulation has to be carried out. The fleet setup label has the vehicle types box where the vehicle parameters, such as the maximum number of miles it can travel, the maximum time it can be on the road, etc., can be defined. More than one vehicle type can be defined in Routepro. All these boxes play an important role in the project to get an optimized routing solution. The implementations of all these properties are discussed in detail in the section to be followed. The other important box is routing, which contains all the routing methods. This box contains all the eight algorithms and their respective parameters that are discussed in the previous chapter. The other boxes need to be accessed when building the routing network and need not be accessed frequently. The other important data structure is the table. The important tables are the sites and orders tables. These provide all the necessary information about every particular site and its order. A site table consists of the site ID, the address of the site, its geocoded coordinates, its site type, and the

pickup and delivery window classes. The orders table consists of the order ID and the number of orders. These are the important boxes and tables that a user encounters frequently during the simulations.

Results and Analysis Boxes: The results that are obtained from the simulation using Routepro are presented in various formats. The important parameters that one is interested in the result are the sequence of the routing, the time window at each site, total miles, unrouted sites, etc. They are presented in dialog boxes, in Excel charts, and also in Gantt charts. The Route summary dialog box in Routepro provides an insight into all the characteristics of each route. The output also presents the route's starting and ending time, the route totals, route costs, performance utilization, and time usage. This box gives a thorough analysis of the routes. The most important information we get after routing is that of the route schedules. This information is necessary for the driver when he sets out on his tour.

4.2.2 INITIAL TESTING FOR THE ROUTING ANALYSIS

Before starting the simulations, it is important to determine the best algorithm for the current routing scenario, since each algorithm is typically best suited for a specific situation. The central aim for the Bon Secours analysis was to transition to a single depot routing operation. The algorithms that are suited and applicable to this scenario from among the various available within the RoutePro tool were: (i) The Nearest Neighbor algorithm, (ii) The Boundary algorithm, and (iii) The Cluster algorithm. Tests were conducted to evaluate these three methods. In the first test, the minimum and maximum angles in both the nearest neighbor and the boundary algorithm were kept at 30 and 120.

The cluster algorithm does not have any angular dependence. All the other parameters were kept identical between the three routing methods. The boundary algorithm came out to be the most efficient. The other factor that was taken into consideration was the amount of memory space used by each algorithm. There was not much difference between the three algorithms in this regard. Another test to gauge the relative merits of the algorithms was conducted by changing the minimum and maximum angles in the nearest neighbor and boundary algorithms to 110 and 180, respectively. The results again underscored the efficiency of the boundary algorithm. Taking all of the above factors into consideration, the present routing and optimization project was, therefore, carried out using the Boundary algorithm.

4.2.3 SIMULATION RESULTS

(a) Baseline simulation: The current routing structure involving two hubs – one at Maryview and the other at DePaul – was analyzed. The mileage currently reported for these routes, originating from the two hubs, served as the baseline for all mileage comparisons. Using the exact delivery and pick-up sequence as currently in place, the total mileage was found using the Mapquest tool and compared against the data currently reported. Results for all the Maryview and DePaul trips are summarized in Table 4.1. For DePaul, only one route was examined in detail, yielding 89 miles against the current report of 152 miles. This amounts to a 41.4 percent savings. For the Maryview hub, Mapquest yielded a total mileage of 151.60. This was slightly greater than the official 132-mile figure. Thus, the analysis clearly highlighted some discrepancies in the current accounting procedures, and brought out the need to carefully re-evaluate the mileage for the current trips. Perhaps some cost savings can result as the mileage is reduced due to

more accurate and stringent accounting of the trips for the current two-hub scenario. The total time for all the trips combined for the current delivery and pick-up scenario was 32.53 hours.

Table 4.1: Baseline Analyses
 i. Maryview Hospital
 ii. DePaul Hospital

Maryview Hospital	Time	Distance Data Given	Mapquest Distance
Trip #1	8:45am– 11:00 am	25 miles	44.9 miles
Trip #2	11:30 am– 2:30 pm	42 miles	40.8 miles
Trip #3	3:00 pm – 6:00 pm	30 miles	32.8 miles
Trip #4	4:00 pm – 6:30 pm	35 miles	33.1 miles

DePaul Hospital	Time	Distance Data Given
Trip #1	9:00 am – 1:45 pm	146 miles
Trip #2	10:30 am - 1:45 pm	155 miles
Trip #3	3:00 pm – 5:45 pm	61 miles
Trip #4	4:00 pm – 6:45 pm	118 miles
Trip #5	4:00 pm – 7:15 pm	46 miles
Trip #6	5:00 pm – 7:30 pm	152 miles

(b) Two-Hub Calculations: An optimized RoutePro-based routing analysis was carried out for the two-hub scenario. Two-hub simulation involves the simulation carried using Maryview as a separate hub and DePaul as another separate hub. It yielded total mileages of 377 and 100 for the DePaul and Maryview trips, respectively. This represents a reduction from the 678 and 132 miles that are the current baseline reports and an overall 41 percent reduction. The details are given in Table 4.2. According to this simulation analysis, a total of two vehicles would be required for the Maryview routes, and four to services the DePaul sites. This is exactly in keeping with the number of vehicles currently being used. Hence, in terms of the fleet requirements, the results are in keeping with the current numbers. Thus, a net gain in mileage is predicted without incurring additional costs to the fleet.

Table 4.2: RoutePro Optimized Two-Hub Results
i. Maryview Hospital
ii. DePaul Hospital

Maryview Hospital	Time	RoutePro Distance	Vehicle Number
Trip #1	8:45am – 10:30 am	18.3 miles	#1
Trip #2	11:30am – 1:30 pm	34.8 miles	#1
Trip #3	4:05 pm – 6:15 pm	27.3 miles	#1
Trip #4	4:40 pm – 6:10 pm	19.6 miles	#2

DePaul Hospital	Time	Distance Data Given	Vehicle Number
Trip #1	9:00 am – 1:25 pm	53.39 miles	#A
Trip #2	10:30 am - 1:15 pm	90.72 miles	#B
Trip #3	3:00 pm – 6:05 pm	33.86 miles	#A
Trip #4	4:00 pm – 6:45 pm	71.06 miles	#B
Trip #5	4:00 pm – 7:40 pm	54.33 miles	#C
Trip #6	5:00 pm – 7:35 pm	73.10 miles	#D

(c) Single-Hub Routing Scenarios: In this scenario, all of the vehicles were required to start and end at the DePaul hospital. Thus, the aim was to eliminate the Maryview hub and operate the delivery based from DePaul. A total of four scenarios were evaluated. In two of them, the total “time-on-the-road” was set at a maximum of three and four hours. These times were chosen as they are roughly in keeping with the actual duration spans of the current trips. The schedules for the pick-up and delivery were chosen to exactly match the preferred times specified by the various doctors’ offices. The aim, in so doing, was to achieve complete customer satisfaction and a high quality-of-service. Results obtained from the RoutePro simulator for the three-hour run are given in Tables 4.3 (Maximum three-Hour Run With Single DePaul Hub and all time constraints met) and the results obtained of four-hour run is given Table 4.4 (Maximum four-Hour Run With Single DePaul Hub and all time constraints met).

As evident from table 4.3, a total of 15 trips are predicted. The total mileage for all the sites is 742, which still is less than the 810-mile baseline level. It shows that a total of 6 vehicles, the same number as currently in operation) would suffice. Three vehicles will be required for the morning runs. This is consistent with the current status. Also, the result is nearly in line with the practical requirement that three drivers should complete morning runs. The three-driver constraint arises from a practical difficulty in being able to hire more drivers at that time of the day. For the most part, three drivers are sufficient. The total time, in hours, for all the trips combined is 32.53. This is slightly higher than the 30.53 hours under the current baseline scheme. This suggests that it would be more cost effective to use full-time employees or to pay on the basis of mileage, rather than hourly.

Table 4.3: Maximum 3-Hour Run With Single DePaul Hub (all time constraints met)

Run	Time	Miles	Vehicles
Run 1:	9:00 - 9:32	14.63	#1
Run 2:	9:00 - 10:16	30.54	#2
Run 3:	9:21 - 12:17	70.85	#3
Run 4:	9:51 - 12:40	83.58	#1
Run 5:	10:52 - 1:49	50.95	#2
Run 6:	11:41 - 2:08	66.15	#4
Run 7:	12:18 - 1:57	35.31	#3
Run 8:	1:11 - 1:33	16.74	#5
Run 9:	2:53 - 3:08	12.55	#5
Run 10:	3:28 - 6:27	65.37	#5
Run 11:	3:41 - 6:29	65.15	#1
Run 12:	4:01 - 6:52	75.66	#2
Run 13:	4:05 - 7:03	62.99	#3
Run 14:	4:15 - 7:01	55.77	#4
Run 15:	4:47 - 6:34	36.08	#6

Table 4.4: Maximum 4-Hour Run With Single DePaul Hub (all time constraints met)

Run	Time	Miles	Vehicles
Run 1:	9:00 - 9:25	14.57	#1
Run 2:	9:00 - 10:38	38.23	#2
Run 3:	9:10 - 12:49	101.60	#3
Run 4:	10:11 - 2:10	84.97	#1
Run 5:	10:48 - 2:10	85.68	#2
Run 6:	11:53 - 1:50	40.61	#4
Run 7:	12:52 - 1:11	13.18	#3
Run 8:	3:17 - 7:06	87.23	#1
Run 9:	3:25 - 7:09	71.66	#2
Run 10:	3:50 - 6:15	44.74	#3
Run 11:	4:08 - 8:01	81.52	#4
Run 12:	5:11 - 7:02	46.56	#5
Run 13:	5:35 - 5:56	17.66	#6

Next, RoutePro simulations were carried out for all the sites with a central DePaul hub based on a maximum four hours allowable time for any trip. Since more time is now allocated for each trip, it becomes possible to serve a larger cluster of sites before returning to the hub. This has two consequences: (a) The total number of

trips is reduced, as short hops are eliminated, and (b) The total mileage drops since the vehicles no longer have to continue returning to DePaul. The total mileage for all the sites for this 4-hour run case was 728, which again is less than the 810-mile baseline level. Details are shown in table 4.4, which predicts a total of 13 trips. The total trip time is 31.43 hours, which is less than the 32.53 hours predicted for the three-hour case by over an hour. A time-chart of the above trips and the possible assignment of vehicles are shown on the following pages. A total of six vehicles, (the same number as currently in operation) are required. However, vehicles five and six are used for very short durations. Vehicle #6, for example, would be required only for about 30 minutes. As before, three vehicles are predicted to carry out the morning runs. This is consistent with the current status and requirement. As with the three-hour maximum trip scenario, a fourth person is required starting at 11:53 am for the 11:53-1:50 pm run.

In order to gauge potential benefits associated with greater optimization, some minor changes in the pick-up and delivery time windows were allowed. Specifically, scheduled constraints for 12 of a total of 198 sites were relaxed by about one hour. RoutePro simulation results with a central DePaul hub based on maximum three-hour trips were obtained. The total mileage for all the sites for this three-hour with some changes in the timing constraints was 694, which is substantially less than the 810-mile baseline level. Details are shown in table 4.5, which predicts a total of 14 trips. The total trip time is 29.96 hours, which is less than the 30.53 hours for the current baseline by about an hour. This clearly shows that significant optimization and cost savings can result from only

Table 4.5: 3-Hour Runs With Single DePaul Hub (12 of 198 time windows not met)

Run	Time	Miles	Vehicle#
Run 1:	9:00 am – 9:15 am	11.35	# 1
Run 2:	9:00 am – 10:11 am	28.05	# 2
Run 3:	9:00 am – 11:58 am	68.65	# 3
Run 4:	9:51 am - 12:40 pm	83.58	# 1
Run 5:	10:52 am – 1:49 pm	50.95	# 2
Run 6:	12:18 pm – 1:57 pm	35.31	# 3
Run 7:	12:35 pm - 3:12 pm	66.37	# 4
Run 8:	1:11 pm – 1:33 pm	16.74	# 1
Run 9:	3:57 pm – 6:57 pm	64.58	# 1
Run 10:	4:01 pm – 6:52 pm	75.66	# 2
Run 11:	4:15 pm – 7:01 pm	55.77	# 3
Run 12:	4:21 pm - 7:12 pm	57.66	# 4
Run 13:	4:51 pm – 7:09 pm	51.22	# 5
Run 14:	5:35 pm – 6:44 pm	28.26	# 6
Total	9:00 am – 7:12 pm	694.15	Six

very minor changes to the delivery and pick-up schedules. This suggests that it might be worthwhile to examine the possibility of making slight changes to the time-windows and the overall schedule, in keeping with the client preferences. Since client satisfaction has to be taken into account, it would not be possible to implement dramatic changes, but at least an effort to determine the possibility for some modifications to the schedules could lead to substantial pay-offs. A time-chart of the above trips and the possible assignment of vehicles are given on the following pages. A total of six vehicles, the number currently in operation, are required. The added advantage of this scenario is that only three vehicles are required to carry out all of the morning runs. This is in contrast to the previous RoutePro predictions which required the use of a fourth vehicle after about 11:30 am.

A final simulation was carried out with nearly no regard to the time windows or delivery schedules. The only constraints were that the trips had to be completed within three hours, and that the daily quota of visits to each doctor's office was being met. Thus, for example, if a site was supposed to be visited twice, once in the afternoon and then in the late evening, RoutePro assigned two visits over the entire ten-hour period (9 am to 7 pm). The simulation package was allowed to fix the actual visit times in order to minimize the mileage, as long as the two-visits-per-day condition was satisfied. The total mileage for this case turned out to be 627 miles with an overall trip time of 28.47 hours as shown in Table 4.6. Only ten trips and five vehicles were necessary.

Table 4.6: 3-Hour Runs With Single DePaul Hub (minimal constraints)

Run Number	Time Interval	Miles Covered	Vehicle Number
#1	9:00am-11: 58am	78.72	#1
#2	9:00am-11: 59am	69.90	#2
#3	9:00am-11: 55am	56.13	#3
#4	9:00am-11: 59am	59.26	#4
#5	9:00am-11:05am	45.06	#5
#6	2:42pm-5: 40pm	64.37	#1
#7	2:53pm-5: 21pm	49.65	#2
#8	2:53pm-5: 49pm	57.78	#3
#9	2:54pm-5: 53pm	82.02	#4
#10	3:10pm-6: 10pm	64.31	#5
TOTAL	9:00am-6: 10pm	627.20	Five

Practically, the above scenario is not very important since the solution, for the most part, will be unacceptable due to conflicts with the doctors' preferred schedule. However, it does in a sense provide for a theoretical limit to the maximum optimization that can be achievable for a three-hour, combined-hub scenario. The mileage of 627 is clearly well below the 810-mile baseline result, and the time of 28.47 hours is also substantially less than the 30.53 hours under current conditions.

4.3 PROBLEM #2: PROCESS FLOW OPTIMIZATION

The objective of this project is to estimate the efficiency and potential productivity enhancement by introducing a buffer stage in Siemens DEKA assembly line. Since there is a cost associated with creating a buffer space, it also becomes essential to determine the optimal buffer size. The complete manufacturing plant is simulated using a discrete event simulation software tool called **Arena**. Arena allows for discrete inputs and takes account of random, stochastic breakdown and failure events in a sequential process flow environment. The simulation software has the features of graphical model construction, input data distribution fitting, output analysis support and animation of the output. Also, Arena can be used for either discrete event simulation or continuous event simulation. This project involves discrete event simulation as discrete parts are being produced and sent to the next processing machine or section based on uniform process cycle times.

4.3.1 PLANT LAYOUT

The manufacturing plant layout for the DEKA fuel injector system indicates that raw materials are being bought from two component sources – the valve body and shell sub-assembly. Individual parts arriving from these two components are processed and are merged to manufacture a final injector. From the shell sub assembly, the parts are processed and are sent to the next stage, i.e., the housing sub assembly. This station checks for housing preparation, welds the assembly via laser processing, loads the coils and furnishes the prepared part for the Housing leak test. Two laser-weld housing units then follow this stage. The processed parts from these stages are then sent to two “Over-mould” machines. The two machines operate in parallel and the process time for these two machines is twice that of other machines.

The injector sub-assembly is the next stage, and this is the stage where the parts from shell sub assembly and also from the valve body are merged. After merging the parts are sent to the final stages of processing. The parts coming out from the valve sub assembly undergo many operations such as loading of the lower screen, seating the load guide, O-ring preparations, wafer and disk preparations and crimping of the valve body. Similarly, the parts coming out from the two over-mould machines undergo corresponding operations. Finally, the two parts are merged in the injector sub assembly stage and operations such as load and invert valve body, orifice orientation, needle armature, etc., are carried out at this stage. After processing from the injector sub assembly stage, the parts pass through other stages such as lift setter, laser weld valve, leak test machine and the final assembly. Final assembly prepares components by installing the plastic shell, the o-rings and a loading adjusting tube. Thus the final part, the fuel injector is produced and is ready for shipping. The plant layout is as shown in the subsequent figure.

The goal of this project is to improve the efficiency of the entire system by incorporating a buffer in the overall assembly line. Buffers are usually introduced in between machines to curtail the adverse effects of “starved” machines during periods of machine breakdowns and failures. The idea in introducing buffers is to preserve an adequate supply of parts for processing by the machines located downstream from the failed unit. This helps keep the production going, and averts the machine idling due to “starvation” of parts. Consequently, in the process the overall operation of the plant can be increased and the effects of machine failures minimized. Though placing a buffer helps increases

Deka iv Automated product process: Figure (1)

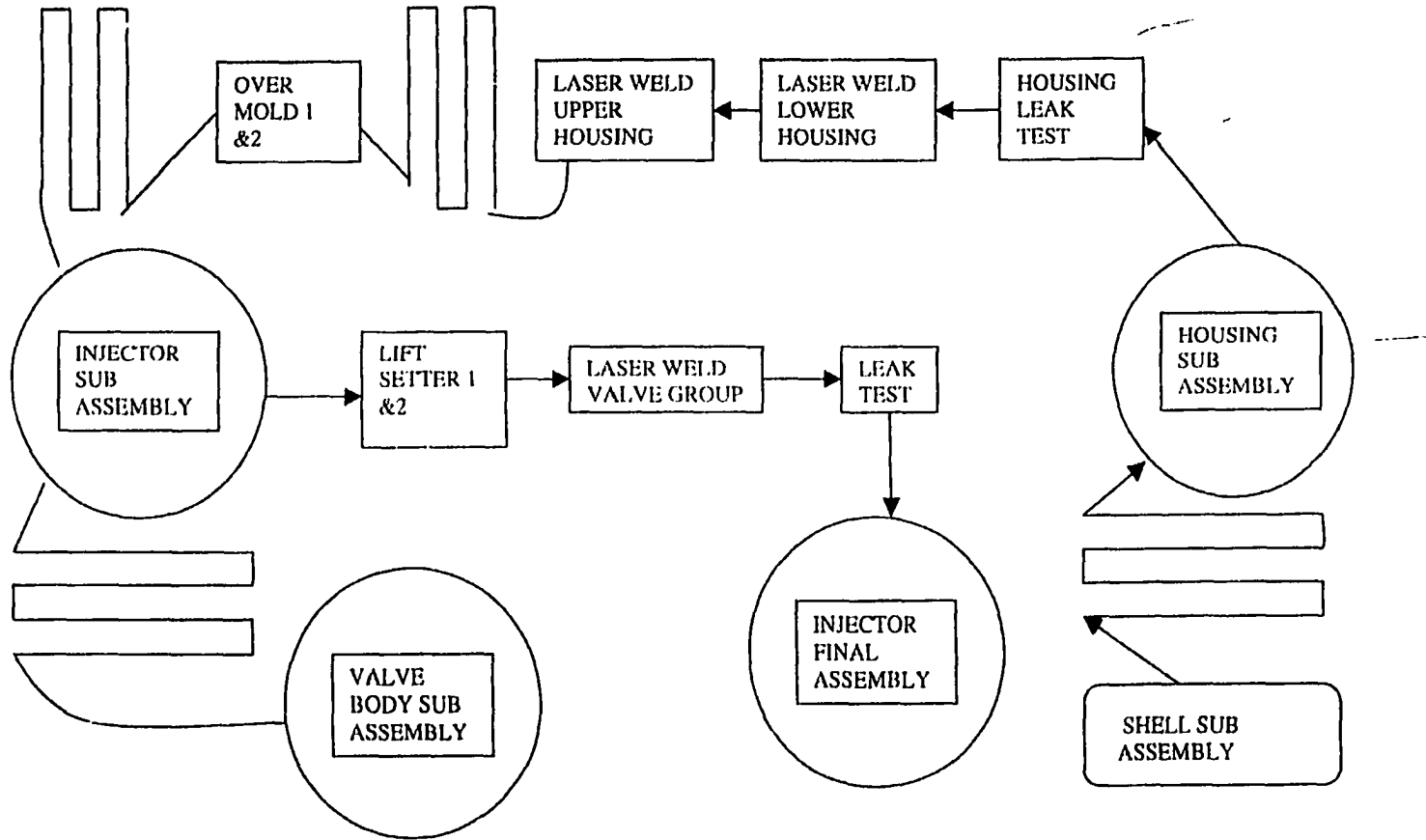


Fig 4.1: Plant layout

the production rate, it comes at the cost of additional space that has to be dedicated for the buffers. Hence, a fixed set-up cost is inherent. For optimization, the goal is to determine the location of such a buffer placement and its optimal size.

4.3.2 DETAILS ON THE SYSTEM DATA

The raw data for the machines involved in the fuel injector valve manufacture was made available by Siemens Automotive, Inc. This data served as the primary resource of the model that has been developed. The data includes a variety of information on all of the machines such as their “up-times,” the “down-times,” the types of faults and failures. Since each event (such as a failure) is stochastic in nature, the data was gathered over a long time period, about a month. The long-term was chosen to provide a better statistical sample that would better represent the stochastic discrete processes. The physical layout yielded a good estimate of the number of pucks that are currently available on each of the connecting conveyor belts. These pucks carry the parts from the one machine to other. Hence, the number of pucks directly determines the number of injector parts that can be placed in the system during production. This information, in the model, is used to designate the “buffer capacity” in-between successive machines. The simulation model was constructed according to the physical location of the machines and the belt sizes. Machines process time provides information about the time that a part takes while passing through that machine. If the machine is without any defects, then it produces the maximum number of parts per hour.

In reality this does not happen, and the output is always lower than the theoretical maximum for two reasons. First, all machines fail at certain times, and remain “off-line”

for a random duration. This halts production, and the overall throughput rate drops. Next, there is also an associated problem of “starving.” During a machine failure all the machines downstream cannot function unless parts stored in buffers or the conveyor space are available for further processing. Even if the conveyors are full, the machines downstream can only operate until they all the available parts are processed. Beyond this point in time, the downstream machines would be “starved,” unless additional supplies from buffers are made available. Thus, placement of buffers helps prolong the production activity, and keeps machines downstream running. Clearly, for best results, the buffers need to be placed immediately after the most failure-prone or inefficient machines in the system. “Blockage” can also arise when a machine breaks down and stops functioning. All the units “upstream” will stop producing parts due to blockage as the buffers and conveyor space upstream fills up. Increasing the belt capacity or placing buffers will alleviate this “blockage” problem as well. The plant layout for the DEKA system shows that there are fourteen machines involved in the production and each machine fails independent of the others. Thus, the machine downtime parameter becomes very important in the number of parts being produced, and in selecting the site for a buffer placement.

Another set of parameters that were obtained for the simulations were the machine efficiencies. If the parts are not processed properly and/or are produced with defects, then the particular part cannot be used for further processes. Such errors directly reduce the total production and the efficiency of the plant.

Buffer capacities:

The conveyor sizes given for this project are summarized below. The following symbols have been used to denote the machines.

H:	Housing
HL:	Housing Leak Test
LHW:	Lower Housing Weld
UHW:	Upper Housing Weld
OM:	Overmold (OM1 and OM2 are in parallel)
VB:	Valve Body Sub-Assembly Dial
IS:	Injector Sub-Assembly Dial
LS:	Lift Set (Stroke1 and Stroke2 are in parallel)
FW:	Final Weld
FL:	Final Leak
F:	Final Assembly Dial

The conveyor capacities for each machine are taken from the data obtained from Siemens. Along with the conveyor capacities another parameter that is also obtained is the machine efficiency. The efficiency for each machine is summarized in the block diagram shown in Fig 4.2.

4.3.3 SIMULATION MODEL

Based on the data given, a model was developed and implemented for analysis. The important parameters that had to be input to the model were: (a) the machine processing

times, (b) the uptime and the downtime of the machine, (c) the machine efficiencies and, (d) the conveyor sizes. The average uptime and the downtimes are calculated using the data available and the following formulae were used.

Calculation of uptime and downtime for each machine:

The given data included the following specifications for each machine:

- (a) Total minutes elapsed for the errors
- (b) Total number of errors
- (c) Total run time for each machine

Based on these, the following quantities were calculated:

- (a) Error minute per errors = Total minutes of failures/ Total no. of failures
- (b) Total minutes per errors = total minutes/errors
- (c) Up time for the machine = (Total minutes per Errors) – (Error minute per errors)
- (d) Downtime for the machine = Error minutes per errors.

The uptime and the downtime calculated using the given data is summarized in the table 4.7 below. The data contains two sets of uptime and down time, one when the data is collected for seven days a week and the other when the data is collected for five days a week (excluding Saturday and Sunday). Both the two sets were analyzed in detail. It

was found that the values for the two sets were fairly close, thus confirming the uniformity of the input data.

The failure rate changes from machine to machine in a stochastic manner. Therefore, this random aspect had to be incorporated into the raw data. A convenient method of doing so is to use probability random distribution functions for the machine uptime and the downtime. The most commonly used distribution functions are the Gaussian/Normal, the exponential and the Poisson distribution functions.

Once the distribution functions are found, they are input into the simulation model and the simulation is run for a considerably long time so that the system reaches a steady state. If only a small run time is allowed for the simulations, then spurious memory effects might persist and the output might not be accurate. Hence, for accuracy and better statistics, large simulation time windows (typically several days) were used. This ensured good, steady state simulation results. The ARENA model was designed so that the simulation represented a “tightly coupled system.” In a tightly coupled system when one machine fails, the previous machine can only operate as long as there is space to store parts on the conveyor or a possible buffer. When the maximum capacity is reached, the previous machine also stops operating until the backlog at the downstream end gets cleared. The excess parts are either removed manually or the upstream machine is fixed and made to operate. This feature was incorporated in the simulation model.

The plant layout shows that the injector parts are not only processed serially but are also attached with different pieces from a parallel stream where the parts from two serially processed machines are used. The machines, where merging occurs are: “Overmold 1,” “Overmold 2,” “Injector final assembly” and the “Valve body sub assembly.”

Fig 4.2: Block diagram of the model with efficiency

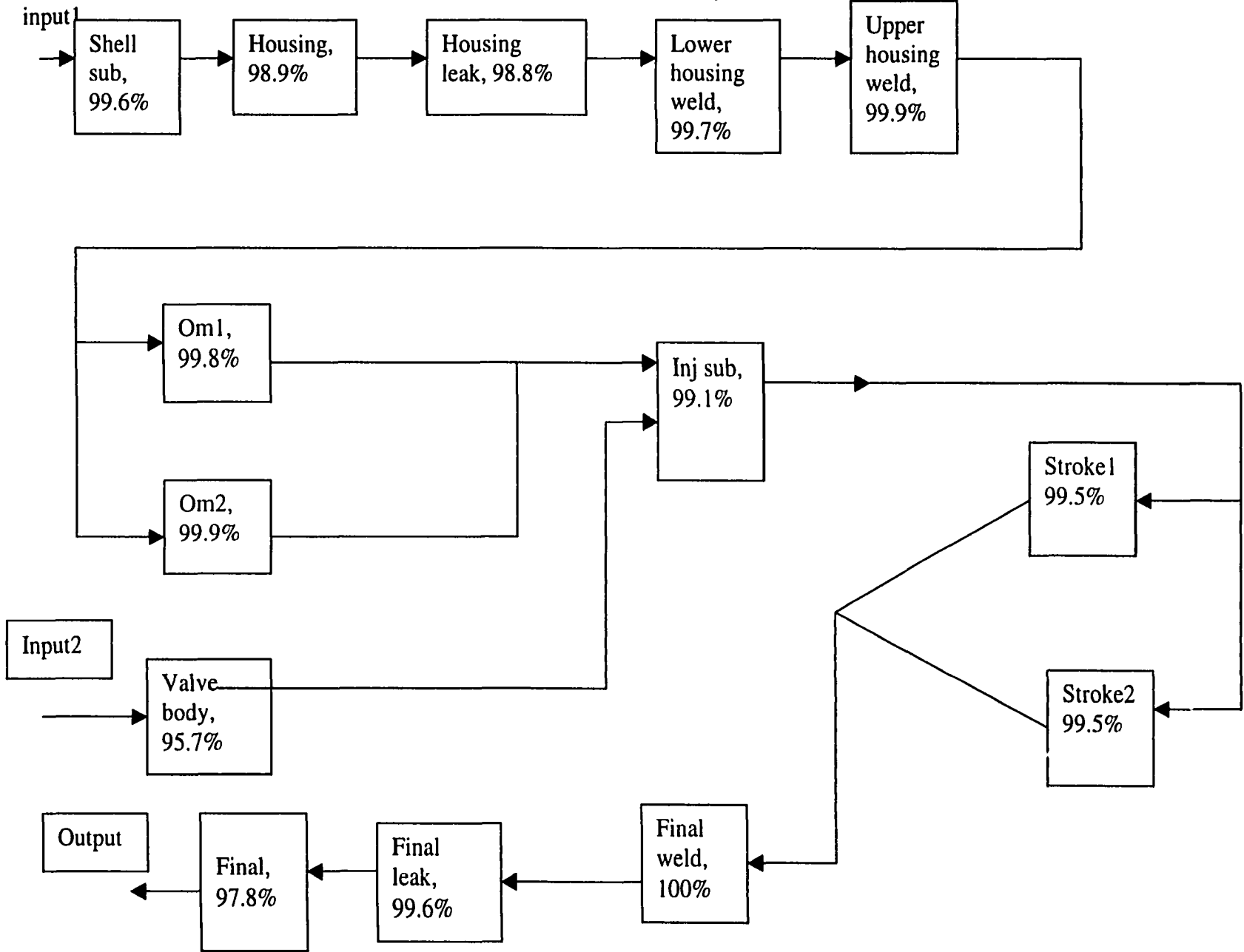


Table 4.7: Uptimes and Downtimes of various machines

	<u>5-DAY DATA</u>		<u>7 – DAY DATA</u>	
	UPTIME	DOWNTIME	UPTIME	DOWNTIME
VALVEBODY	7.319689	1.791511	7.376691	1.761957
HOUSING	6.345528	1.183754	6.536	1.2318
HOUSING LEAK	8.48421	5.074681	8.6026	4.987
LHW	10.194	6.437	14.31827	5.7254
UHW	9.32028	1.367951	9.282	1.962
OM1	21.57043	4.909717	29.12054	4.987646
OM2	23.5099	12.55367	26.96546	18.20023
INJSUB	8.095771	1.482543	8.160959	1.470544
STROKE1	7.744804	1.049314	7.713341	1.013208
STROKE2	11.64305	1.582757	11.618	6.8528
FINAL WELD	10.47838	1.021623	10.518	0.9170207
FINAL LEAK	10.83845	0.566776	10.7555	0.56851
FINAL	8.512898	1.347626	7.759224	1.221668

The parts arriving from the Overmold 1 and Overmold 2 sections wait in the queue of the Injector sub assembly. Similarly, the parts from the Valve sub- assembly also wait in a second queue. If the parts are not available in any queues, proper feeding cannot take place at the injector sub-assembly machine. This feature has been implemented in the simulation model.

4.3.4 SIMULATION RESULTS

Simulation was conducted for the two following scenarios. One (Scenario A) was a three-second cycle time, which corresponds to the present production time. The other (Scenario B) 2.5-cycle time was also used to predict the productivity if the individual machines could be made to operate faster.

Scenario A (3 seconds processing time for each machine):

The processing time for each machine was set at three seconds and different operating scenarios were tested. In each case, the entire production system was simulated for a sufficiently long time and the number of parts produced per hour was determined. On the input side, a three-second feed time was always used. At this rate and under ideal conditions, i.e., no downtime and 100% efficient system, the number parts produced at the output must equal the number of parts given at the input. Thus, under ideal conditions an output rate of 1200 parts/hour is the theoretical maximum. But in reality machine failures, line blockages and starved units will reduce the output rate below the 1200 parts/hour limit. For the data given, the more realistic, non-ideal situation predicted 764 parts/hour. This production level is close to the output levels currently being realized at

Siemens. Thus, this result provides a baseline for comparison and validates our simulation model.

Next, with the three-second machine cycle, simulations were run to obtain the productivity with the entire production system broken into two separate sections after the Overmold unit. Conceptually, this amounts to the placement of an infinite buffer after the Overmold unit. The choice of this placement was guided by the desire at Siemens to evaluate such a scenario. Besides, the Overmold machines are most prone to breakdown. Hence, it is logical to place a buffer after this stage to curtail starvation downstream. Thus, the upper chain was simulated separately, and the lower chain including the merging at the Injector Sub assembly was simulated separately. Infinite availability of parts was assumed at the beginning of both sections. The number of parts being produced per hour by both the stages was determined to be 886 and 957, respectively.

Scenario B (2.5 seconds processing time for each machine):

The same set of data for the various cases mentioned above was obtained based on a 2.5 seconds processing time for each machine. All the other parameters remained the same as in the three-second case. Due to the faster speed of the machines, the output rate increased as expected. For the entire system, an output of 886 parts/hour was predicted. This is a 16 % increase over the baseline value. Simulations for the upper section alone produced 1037 parts/hour, a 17 % increase. Finally, 1143 parts/hour resulted from simulations of the lower section, which is a 19.4 % increase.

The above results for different scenarios are presented below in a tabular form.

(A) 3 second cycle time:

Maximum parts that can be produced are 1200 parts/hour.

Case	Number of parts produced per hour
Simulation result of the entire system	764
Simulation result of 1st chain	886
Simulation result of 2nd chain	957

(B) 2.5 second cycle time:

Maximum parts that can be produced are 1440 parts/hour.

Case	Number of parts produced per hour
Simulation result of the entire system	886
Simulation result of 1st chain	1037
Simulation result of 2nd chain	1143

Clearly, in terms on increasing the productivity and profit margins, it would be best to work with a 2.5-second machine. In reality, however, improving the speed could come at the expense of a loss in efficiency or higher occurrence of machine

failures. These potential changes in the data were not taken into account here due to a lack of any tangible values.

CHAPTER V

CONCLUSIONS AND FUTURE RECOMMENDATION

5.1 INTRODUCTION

Vehicle routing problems have been under study for over two decades, and newer computer-aided simulation software tools are been released to unravel this problem and even provide optimized solutions. Routepro has been one such product that has helped to solve the routing problem in an efficient way. It is sophisticated, and takes into account various minute details that some other simulation tools fail to consider. The efficiency of this software was used to get an optimized routing solution in the context of a network of doctors affiliated with a central hospital facility. Interactions to and from the doctors' sites and the hospital were necessary for purposes of transferring patient specimens for analysis and testing. Different scenarios were studied and based on the solution obtained from the software tool; various recommendations to the hospital were obtained. The solutions and associated savings in mileage obtained by using the Routepro results are summarized and discussed in the following sections.

Another simulation tool under discussion in this thesis research work was Arena. This tool was used in solving a production flow problem for a manufacturing plant producing fuel injectors for cars and trucks. This tool contained graphical interfaces that help the user visualize the various details of the plant. Also, the tool was user friendly and helped in solving the production flow problem. The thesis primarily concentrated on providing justification of the use of buffer at various locations of the assembly line of a manufacturing plant. An attempt to optimize the buffer size at the different locations was

made to increase the overall manufacturing throughput across the assembly line. Assessment of the buffers and quantitative evaluations of the benefits were made. The details of the results obtained from the software tool to optimize the buffer size and the various time windows used to estimate the output are discussed in the sections to follow.

5.2 RESULT SUMMARY FOR THE ROUTING PROBLEM

Simulation for various scenarios was carried out and different algorithms were used to analyze the problem in detail. The algorithm that yielded the best solution was selected and the results obtained from that particular algorithm used. Amongst the various algorithms that were used, invariably the solution obtained from the Nearest Pass algorithm was found to be the most optimized. A common testing strategy was employed while evaluating different algorithms; thus, the relative efficiency of various algorithms was analyzed. The results obtained from the simulation and their salient features are given below. The different scenarios were discussed in the previous chapter, as were the results from the scenarios.

- (i) There appears to be some mileage accounting problem with the current DePaul routes. A somewhat lower mileage should result, thereby providing savings. This was clearly evident from the base run simulations that were carried out, and yielded lower mileage.
- (ii) Time windows are very important for successful optimization. Allowing larger time windows facilitates greater reductions in the mileage and overall

times. Some minor time window alteration for each doctor was done and the simulations were carried out to yield large changes in the total mileage.

- (iii) Longer allowable trip times lower the mileage and total time span as expected. This is done by varying the number of hours a vehicle could be used for continuous running. Although larger trip times give better mileage, it is not practically feasible, since the drivers are not able to physically drive the vehicle continuously for long periods without a break. Also, in this case, since the vehicles were used to transfer human specimen samples, there was a need to bring them into controlled laboratory environments in a timely manner. Failure to do so would jeopardize their integrity and lead to sample degradation. So, an optimal number was chosen for the number of hours the vehicles could be on the road. This number was chosen in such a way that it closely matched the time spans currently being used.
- (iv) A summary of the simulation results under the various scenarios is given in the table below. The various scenarios include the baseline run, changing the number of allowable trips, and also, by changing the time windows for pick-up and delivery visit to doctors' offices. The solution also provided details on the number of vehicle used in each trip, total miles traveled and the total time for completing the route. The decision to select the best solution from the various scenarios was given to the management of the hospital. Based on the above summarizing table, it appears that the best option would be the 4-hour scenario with all constraints satisfied. In terms of the mileage, a 10.12 % reduction is projected.

Scenario	Total Time	Total Mileage	Comments
Baseline/Current	30.53 hours	810 miles	6 vehicles; 3 morning drivers
Depaul, 3-hour (all constrained)	32.53 hours	742 miles	Savings in mileage at expense of time. 6 vehicles; 3 morning drivers, 1 after 11:40 am.
Depaul, 4-hour (all constrained)	31.43 hours	728 miles	Savings in mileage, but not in time. 6 vehicles; 3 morning drivers, 1 after 11:53 am. Sixth vehicle used only for 25 minutes.
Depaul, 3-hour (nearly all constraint)	29.96 hours	694 miles	Savings in mileage, and in time. 6 vehicles; 3 morning drivers. Sixth vehicle used only for 70 minutes.
Depaul, 3-hour (minimal constraints)	28.47 hours	627 miles	Savings in mileage, and in time. 5 vehicles; 5 morning drivers.

Table 5.1: Summary of results of the routing problem.

- (v) Also, sufficient numerical evidence has been presented that justifies a systematic study into the scheduled delivery and pick-up times. Possible relaxation of at least a few delivery and pick-up times would lead to a large reductions in mileage and total trip hours.

5.3 RESULTS OF THE PRODUCTION LINE PROBLEM

The effects of introducing buffers and the determining the optimal buffer size in the production line was the primary objective of this second thesis task. The simulation solution helped in obtaining the buffer size, thereby increasing the production throughput and the related profit margin for the production plant. The necessary data required for the simulation model was obtained and analyzed. Using the data an Arena model was built and different scenarios were tested. The results obtained from the simulation showed that by increasing the processing time of each machine and also by increasing the buffer size, the number of parts produced per hour could be increased. This should lead to increased profits. A strong case was, therefore, made for introducing a buffer for productivity enhancement.

From the results given in the last chapter, the number of parts per hour for the entire plant was 1200 parts/hour if all machines uniformly took three seconds to process a part, and *operated without any failures*. However, with failures, 764 part/hour would result for the three-second processing rate per part. The simulation result of 764 parts/hour was found to be in close agreement with to the actual production numbers per hour for the DEKA system. Also, a substantial productivity increase was predicted by having an infinite buffer placed before the merging section of the assembly line. This increase translated

into: $(957-764)/764 = 25.26\%$ variation. If the lower section in the plant after the merger were to be in operation for five days a week, then the initial section (prior to the infinite buffer) would have to run for: $5 \times (957/886) = 5.4$ days. The actual buffer capacity would be given by the differential between the two. This would work out to: $(5.4 \text{ days} - 5 \text{ days}) \times 24 \text{ hours/day} \times 886 \text{ parts/hour} = 8520$ parts. The weekly number of parts produced (roughly in this 5.4 days/week span) would be: $5.4 \times 24 \times 886 = 114,825$ parts.

If, instead, a decision was made to keep the initial section running only for a maximum of 5-days/week, then the lower section would need to be operational only for $(886/957) \times 5 = 4.63$ days. This would save operating costs by reducing the “man-hours” per week. The buffer capacity would then be: $(5 \text{ days} - 4.63 \text{ days}) \times 24 \text{ hours/day} \times 886 \text{ parts/hour} = 7868$ parts. The weekly number of parts produced roughly in a 5-days/week span would be: $5 \times 24 \times 886 = 106,320$ parts. Though slightly less than the 114,825 result above, there would be savings by shutting down the downstream section for $(5.0 - 4.63) = 0.37$ days ~ 8.9 hours.

It was predicted that running the machines of the DEKA system on a 2.5-second cycle would lead to increased productivity. Transition from a 3.0-second to a 2.5-second cycle translated into a 20 % increase in productivity under ideal conditions. However, the percentage increases obtained from the actual simulations of the non-ideal system were: 16 %, 17 % and 19.4 % for the complete system, the upper section and lower sections, respectively. This is not surprising since the throughput can scale as the machine speed only if other factors do not change. However, effects such as “starved” and “blocked”

units will reduce effectiveness of machine-cycle increases. Since machines in the lower section have the least breakdowns, the starvation and blocking phenomena is the least. Consequently, the speed up in going to a 2.5-cycle time is the most and approaches the theoretical gain of 20 %.

Similar comparisons of the numerical data as done for 3.0-second cycle indicated the following. With the addition of an infinite buffer, the productivity would increase from 886 parts/hour to a maximum of 1143 parts/hour. This represents a 29 % increase. If the lower section were to be in operation for five days a week, then the initial section would have to run for: $5 \times (1143/1037) = 5.51$ days. The actual buffer capacity would be given by the differential between the two. This would work out to: $(5.51 \text{ days} - 5 \text{ days}) \times 24 \text{ hours/day} \times 1037 \text{ parts/hour} = 12,720 \text{ parts}$. The weekly number of parts produced (roughly in this 5.51 days/week span) would be: $5.51 \times 24 \times 1037 = 137,132 \text{ parts}$. If instead a decision was made to keep the initial section running only for a maximum of 5-days/week, then the lower section would need to be operational only for $(1037/1143) \times 5 = 4.5363$ days. This would save operating costs by reducing the “man-hours” per week. The buffer capacity would then be: $(5 \text{ days} - 4.5363 \text{ days}) \times 24 \text{ hours/day} \times 1037 \text{ parts/hour} = 11,540 \text{ parts}$. The weekly number of parts produced roughly in a 5-days/week span would be: $5 \times 24 \times 1037 = 124,440 \text{ parts}$. Though slightly less than the 137,132 result above, there would be savings by shutting down the downstream section for $(5.0 - 4.5363) = 0.4637 \text{ days} \sim 11.12 \text{ hours}$.

Clearly, in terms on increasing the productivity and profit margins, it would be best to work with a 2.5-second machine. In reality, however, improving the speed could come at

the expense of a loss in efficiency or higher occurrence of machine failures. These potential changes in the data were not taken into account here due to a lack of any tangible data on the speed dependence of efficiency.

5.4 FUTURE WORK

A list of recommendation for the future work is given below. For the vehicle routing problem involving optimization of the routes, the following are recommended for possible future work.

- (i) Conduct systematic surveys of the doctors' preferences in an attempt to expand the allowable time windows for pick-up and delivery. Since better optimization, lower total mileage and overall trip times result, as the time constraints are made less rigid, there may be economic benefits to using a slightly different set of schedules. However, the tolerance level for such changes has to be determined from direct questionnaires sent to the various sites for their input.
- (ii) The present study focused on a single-depot situation. As a natural extension, one could undertake a study for single-depot, but multiple distribution/pickup nodes. This would lead to the idea of a "local area networks" for "near neighbor" servicing. This could minimize trips all the way back to the central DePaul hub each time.
- (iii) In the present case, the data used for the input parameters was taken to be fixed and time invariant. In actual practice, there are fluctuations on a daily basis. A more realistic analysis could, therefore, include such statistical aspects. At the

very least, a weekday and a weekend grouping could be carried out. It might then lead to the conclusion that multiple routing strategies have to be used between weekdays and weekends.

- (iv) Finally, similar analysis could be carried out to ascertain the viability of expanding the transportation network to new areas and cities.
- (v) The geographical database and the graphical interface capabilities of the software helped to gain a better understanding of the routes and give a good visual effect from both the users' point of view and the analysts' standpoint. However, for future analysis, this database may need to be expanded to incorporate changes in the local geography in response to on-going regional developments.

For the project involving discrete event simulation for the continuous production of fuel injectors, the following are recommended as possible tasks for the future.

- (i) Some additional data could be included in the model such as the schedules and time required for machine maintenance, stoppages due to wear and tear of the machines, and failures due to various other reasons. Minute monitoring of the machines could help in collecting this data. Addition of all these data could alter the solution.
- (ii) Different distribution curves could be tried for various machines and a better solution might be obtained.

(iii) Sections that affect the overall performance most strongly could be established, and schemes for improvements focused on these specific sections for a better solution. A possibility would then be to incorporate several buffers of varying sizes at various locations for a complete optimization.

(iv) Parallel processing units could be introduced for enhanced production rates, and their impact studied.

REFERENCES

- [1] Bertsimas, Dimitris J and D. Simchi-Levi., 1996, A New Generation of VehicleRouting Research: Robust Algorithms, Addressing Uncertainty, *Operations Research*, **44**, pp. 286 – 304.
- [2] Bentley, J and Saxe, J., 1980, An Analysis of Two Heuristics for the Euclidean Traveling Salesman Problem, *Proc. Of the 18th Annual Allerton Conference on Communication, Control and Computing*, 41-49.
- [3] Laporte, G., 1998, Recent Advances in Routing Algorithms, *NATO ASI F Series Computer and Systems Sciences*, **166**, 183-202.
- [4] Ball, M., and Magazine, M., 1981, The Design and Analysis of Heuristics, *Networks*, **11**, 215-219.
- [5] Akl, S., 1978, An Analysis of Various Aspects of the Traveling Salesman Problem, *Ph.D. Thesis*, McGill University.
- [6] Maffei, R., 1965, Modern Methods for Local Delivery Route Design, *Journal of Marketing*, **29**, 13-18.
- [7] Achutan, N.R., Cacetta, L., and Hill, S.P., 1997, On the Vehicle Routing Problem, *NonLinear Analysis, Theory, Methods and Applications*, **7**, 4277-4288.
- [8] Gilbert Laporte., 1992, The Vehicle Routing Problem: An overview of exact and approximate algorithms, *European Journal of Operational Research*, **59**, 345-358.
- [9] Jean-Francois Cordeau, Michel Gendreau and Gilbert Laporte, 1997, A Tabu Search Heuristic for Periodic and Multi-Depot vehicle routing problems, *Networks*, **30**, 105-110.

- [10] Marius M. Solomon, 1987, Algorithms for the Vehicle Routing Problem and Scheduling Problems with Time Window Constraints, *Operations Research* , **35**, 254-265.
- [11] Barrie M. Baker and Janice Sheasby, 1999, Extensions to the generalized assignment heuristic for vehicle routing, *European Journal of Operational Research*, **119**, 147-157.
- [12] Oliver, I.M., Smith, D.J. and Holland, J.R.C., 1987, A study of permutation crossover operators on the traveling salesman problem, *Genetic Algorithms and their Applications*, 224-225.
- [13] Khloudova, M.V, 1999, Real-time Scheduling in a Stochastic Environment, *Production of the International Society for Optical Engineering*, **4064**, 259-263.
- [14] Blumenfeld, D. E., 1990, A Simple Formula for Estimating Throughput of Serial Production Lines with Variable Processing Times and Limited Buffer Capacity, *Int. J. Production Research*, **28**, 1163-1182.
- [15] Johri, P. K., 1987, A Linear Programming Approach to capacity Estimation of Automated Production Lines with Finite Buffers, *Int. J. Production Research*, **25**, 851-866.
- [16] Buzacott, J. A., 1967, Automatic Transfer Lines with Buffer Stocks, *Int. J. Production Research*, **5**, 183-200.
- [17] Kalir, A. and Arzi, Y., 1997, Automated Production Line Design with Flexible Unreliable Machines for Profit Maximization, *Int. J. Production Research*, **35**, 1651-1664.

- [18] Ghosh, S. and Gargon, R. J., 1989, A Comprehensive Literature Review and Analysis of the Design, Balancing, and Scheduling of Assembly Systems, *Int. J. Production Research*, **27**, 637-670.
- [19] Kalir, A. and Arzi, Y., 1998, Optimal Design Of Flexible Production Lines With Unreliable Machines And Infinite Buffers, *IIE Trans.*, **30**, 391-400.
- [20] Malkooti, B. B., 1994, Assembly Line balancing with Buffers by Multiple Criteria Optimization, *Int. J. Production Research*, **32**, 2159-2178.
- [21] Martin, G. E., 1994, Optimal Design of Production Lines, *Int. J. Production Research*, **32**, 989-1000.
- [22] Gershwin, S. B., 1987, An Efficient Decomposition Method for Approximate Evaluation of Tandem Queues with Finite Storage Space and Blocking, *Operations Research*, **35**, 291-306.
- [23] Dallery, Y., David, R., and Xie, X., 1989, Approximate Analysis of Transfer Lines with Unreliable Machines and Finite Buffers, *IEEE Trans. Automatic Controls*, **34**, 943-953.
- [24] Hunt, G. C., 1956, Sequential Arrays of Waiting Lines, *Operations Research*, **4**, 674-683.
- [25] Hillier, F. S., and Boling, R. W., 1967, Finite Queues in Series with Exponential or Erlang Service Times – a Numerical Approach, *Operations Research*, **15**, 286-303.
- [26] Basu, R. N., 1977, The Interstage Buffer Storage Capacity of Non-Powered Assembly Lines: A Simple Mathematical Approach, *Int. J. Production Research*, **15**, 365-383.

- [27] Muth, E. J., 1977, Numerical Methods Applicable to a Production Line with Stochastic Servers, *TIMS Studies in the Management Sciences*, **7**, 143-159.
- [28] Muth, E. J., 1987, An Update on Analytical Models of Serial Transfer Lines, *Research Report*, **87-15**, (University of Florida, Gainesville, FL).
- [29] Fugazza, M., 1993, A Software system for runoff and routing modeling in Drainage Catchments, *Proc of the International conference on Urban Drainage* **1**, 230-236.
- [30] Robinson, S., and Bhatia, V., 1995, Secrets of Successful Simulation Projects, edited by Alexopoulos, C. Kang, K., and Goldsman, D., *Proceedings of the 1995 Winter Simulation Conference*, 15-18.
- [31] Banks, J., 1996, Output Analysis Capabilities of Simulation Software, *Simulation*, **66**, 23-30.
- [32] Harrel, C., Ghosh, B. K., and Bowden, R., 2000, *Simulation Using ProModel*, (McGraw-Hill, New York.)
- [33] Ushimura, K., and Takahashi, H., 1999, Vehicle Routing and Scheduling for Hierarchical Public Transportation System, *ISATA Proceedings*, **32**, 119-126.
- [34] Fleishmann, B., Van Nunen, J. E., Grazia, M., and Speranza, S. P., 1988, Advances in Distribution Logistics, *Lecture Notes in Economics and Mathematical Systems* , 460-474.
- [35] Madhu, K., and Saxena, S., 1998, Parallel Algorithms for Vehicle Routing Problems, *International Conference on High Performance Computing*, **5**, 171-178.

- [36] Thangiah, O., Vinayagamoorthy, M., and Sun, T., 1993, Algorithms for the Vehicle Routing Problem with Time Deadlines, *American Journal of Mathematical and Management Sciences* , **13**, 323-355.
- [37] Laporte, G., 1992, The Vehicle Routing Problem: An Overview of Exact and Approximate algorithms, *European Journal of Operational Research*, **59**, 345-358.
- [38] Agin, N., and Cullen, D., 1975, An Algorithm for Transportation Routing and Vehicle Loading, *Logistics*, **1-20**, North Holland, Amsterdam.
- [39] Hamacher, A., and Moll, C., 1992, A New Heuristic for Vehicle Routing with Narrow Time Windows, *European Journal of Operational Research*, **59**, 365-378.
- [40] Unveren, A., and Acan, A., 1999, A Powerful Combination of Genetic and Greedy Algorithms for Over-the-Cell Channel Routing Problems, *Intelligent Engineering Systems Through Artificial Neural Networks*, **9**, 369-374.
- [41] Nygard, K.E., Magel, R., and Yang, C., 1992, Conversational Statistical Computing for Routing Algorithms, *Applied Computing Symposium*, **2**, 1150-1158.
- [42] RoutePro Reference Guide, *CAPS Logistics Inc.*
- [43] Kelton, D., Sadowski, R., and Sadowski, D., 1998, *Simulation with Arena*, (McGraw-Hill, N. York.)

CURRICULUM VITA
for
SATISH RAVICHANDRAN

DEGREES:

Master of Science (Electrical Engineering), Old Dominion University, Norfolk,
Virginia, August 2001

Bachelor of Engineering (Electronics and Communication Engineering), Bharathiar
University, Coimbatore, India, May 1999

CONSULTING/PART TIME EMPLOYMENT:

Virginia Modeling, Analysis and Simulation Center, Graduate Research Assistant,
January – December 2000.

Old Dominion University Department of Electrical Computer Engineering, Graduate
Teaching Assistant, Fall 2000.

COURSES TAUGHT:

ECE 286 - Computer Aided Tools in Electrical and Computer Engineering, Fall
2000.