Using Genetic Algorithms for the Single Allocation Hub Location Problem

Mohammad Naeem

Submitted in partial fulfillment

of the requirements for the degree of

Master of Science

Faculty of Computer Science, Brock University St. Catherines, Ontario

Contents

List of Figures

List of Tables

Abstract

Hub location problem is an NP-hard problem that frequently arises in the design of transportation and distribution systems, postal delivery networks, and airline passenger flow. This work focuses on the Single Allocation Hub Location Problem (SAHLP). Genetic Algorithms (GAs) for the capacitated and uncapacitated variants of the SAHLP based on new chromosome representations and crossover operators are explored. The GAs is tested on two well-known sets of real-world problems with up to 200 nodes. The obtained results are very promising. For most of the test problems the GA obtains improved or best-known solutions and the computational time remains low. The proposed GAs can easily be extended to other variants of location problems arising in network design planning in transportation systems.

Acknowledgements

I express my gratitude to my advisor, Professor Beatrice Ombuki-Berman whose valuable guidance, support, and motivation made this work possible and who helped me learn academic writing. I thank Professor Brian Ross and Professor Vladimir Wojcik, who were always there to guide me in this work.

My special thanks go to Dr. Andreas T. Ernst, CSIRO, Australia, one of the pioneers in SAHLP research, for sending me the large-sized AP data and helping me with some of the conceptual points in the Single Allocation Hub Location Problem. Professor Martin O'Kelley, Ohio State University, USA, a pioneer of hub location research, was kind enough to send me the CAB data and his papers on how to use it for which I express my gratitude to him. Thanks are also due to Ivan Contreras of Technical University Catalonia, Spain for sending me some new AP data. Furthermore, I thank M. Ermisb, Marmara University, Turkey and Marcus Randal, Bond University, Australia for help and information.

Last, my thanks go to Tanaby Zibamanzar Mofrad and Earl Brendan Foxwell, M. Sc. students at Brock University for useful discussion during this work. I also thank my wife and children for their assistance and patience.

Chapter 1

Introduction

Hub Location Problems (HLP) are classical combinatorial optimization problems that arise in telecommunication and transportation networks where nodes send and receive commodities (i.e., data transmissions, passengers, express packages, mail, etc.) through special facilities or transhipment points called hubs. Hubs consolidate flows from origin nodes and re-route them to destination nodes sometimes via other hubs. The sending and receiving nodes in such networks are called spokes. The networks are called hub-spoke networks. The assumption in hub-spoke networks is that, hubs are fully-connected through low-cost high-volume pathways that allow a discount factor to be applied to the transportation cost of the flow between a given hub pair. Another assumption in these networks is that, all the internodal flow takes place through at least one hub and at most two. Broadly, the hub location problem (HLP) is concerned with locating hubs on the network and allocating spokes to the hubs so as to minimize total flow cost subject to the above assumptions.

Hub-and-spoke networks have application in many areas. Common examples include passenger airlines [14, 15, 16], express package delivery firms [17], message delivery networks [18], trucking industry [21], telecommunication systems [23], supply-chain of chain stores such as Wallmart [20], and many other areas. Many studies have indicated that the implementation of hub-and-spoke network has improved the performance of the distribution system. Due to their multiple applications and economic value, HLPs have received much attention in literature.

Hub location problem has many varieties according to the constraints and decision variables involved such as the way of selecting the number of hubs to be located, the way the spokes are assigned to hubs, the existence of capacity limits on hubs, etc. A comprehensive survey on Hub Location Problems (HLPs) and their classification can be found in Kara *et. al.* [10]. In this thesis, a variant of the

HLP i.e., the Single Allocation Hub Location Problem (SAHLP), is considered.

In the Single Allocation Hub Location Problem (SAHLP), a spoke is allocated to exactly one hub and the number of hubs to be used is not known in advance. Furthermore, hubs are capacitated or uncapacitated. Capacitated hubs can handle limited internodal flow whereas uncapacitated hubs can manage any amount of flow. Corresponding to these hub types, two variants of SAHLP exist; the Capacitated Single Allocation Hub Location Problem (CSAHLP) with capacity limits on hubs; the Un-capacitated Single Allocation Hub Location Problem (USAHLP) involving hubs with unlimited capacities. An example of the capacitated SAHLP application is in postal delivery systems in which a sorting center (or hub) sorts and consolidates mail arriving from different postal districts and re-route it to the destination [1] usually through other centers. The sorting centers in such systems have capacities i.e., they can handle a maximum amount of mail flows from origin-destination points. Example of the application of the uncapacitated SAHLP is the air transportation networks [24].

The SAHLP is an NP-hard problem [10]. Additionally, in SAHLP, the number of hubs is not known *a priori* and the single assignment constraint i.e., a given spoke must be allocated to only one hub, holds. Furthermore, in the capacitated version of the SAHLP, hubs have capacity limits to handle the flow between nodes. This makes the SAHLP problem more challenging. Due to their usefulness and economic importance, both the capacitated and uncapacitated versions of SAHLP have received a good amount of research attention and exact and heuristic methods have been proposed to tackle them. Some of these methods include a quadratic integer programming formulation [24] and its linearization [10] , Genetic Algorithm (GA) [3], hybrid heuristic combining GA and Tabu Search [4], and a Simulated Annealing (SA) and Tabu Search (TS) based hybrid solution method [11] for the uncapacitated SAHLP. For the capacitated SAHLP, a branch and bound technique combined with two heuristic procedures [1], a GA [13], an Ant Colony Optimization (ACO) based solution approach [2] have been proposed. These approaches have been able to solve the SAHLP problems of up to 50 nodes to optimality [1]. Modest results, however, have been achieved for larger SAHLP problems, which continue to be computationally intractable.

Genetic algorithms are biologically inspired heuristic that evolves improved solutions to a computationally hard problem through the process of selection and recombination. They have been applied, with a good degree of success, to many combinatorial optimization problems including the SAHLP as mentioned above. Although, the GAs [3][4][13] for the SAHLP were generally successful, they couldn't find known-best solutions to some of the SAHLP problems, especially the medium and large-sized capacitated SAHLP problems. Moreover, to the best of the author's knowledge, only one GA has been proposed to the capacitated SAHLP so far. Furthermore, the existing GA approaches to the SAHLP employ conventional GA techniques like one-point crossover etc., which may not be very effective for the complex solution structure of the SAHLP. There is thus room for further GA-work on the SAHLP. Solutions for the large-sized SAHLP problems can be improved by employing different GA techniques e.g. problem-specific crossovers and solution representations that suit the structure of the problem.

In the GA in this work a,GA on problem-specific crossovers has been proposed to the SAHLP i.e., the *Double-Cluster Exchange Crossover(DCEC)* and *Multi-Cluster Exchange Crossovers (MCEC)*. The approach adopted in these crossovers treat clusters i.e., a hub with associated spokes, as units of gene exchange between the mating parents instead of individual nodes as in the existing GAs for the SAHLP. The crossovers also perform partial handling of the hub capacity constraint in the capacitated SAHLP. This approach to constraint handling has been observed to have good influence on the overall performance of the GA. A third crossover used in the proposed GA is the *Best Cost Routing Crossover (BCRC)* [23]. The Best Cost Routing Crossover (BCRC) has been used in routing problems with good results [23].

Two solution encoding schemes, the *List-based* and *Set-based* encodings, have been employed in the proposed GA. The list-based encoding uses an allocation array like structure to represent the location of hubs in the network and allocation of spokes to hubs. The Set-based scheme employs sets to encode location-allocation information in the hub-spoke network. Besides, the GA incorporates an efficient constraint handling procedure to handle the capacity constraint on hubs in the capacitated SAHLP. The technique attempts to preserve or enhance the fitness of an infeasible solution through a process of careful re-assignment of nodes while adjusting overflow in a hub.

Three mutation operators i.e., the *shift mutation*, the *swap mutation*, and the *replace hub mutation* have been employed in the proposed GA. The shift and swap mutations have been used in previous GA studies on SAHLP and have found to be effective. The *replace hub mutation* has been introduced to change hub location during the re-production and so to preserve population diversity.

Three versions of the proposed GA i.e., GA-1, GA-2, and GA-3, each based on one of the aforementioned crossovers, have been implemented. The computational performance of the GAs has been investigated through extensive experimentation with two sets of standard benchmark problems derived from real-world applications. The CAB (Civil Aviation Board) data is based on air-traffic flows between 25 cities of US and has been extensively used as benchmark for the uncapacitated hub location problems. It includes problem instances of up to 25 nodes. The Australian Post (AP) data has been derived from a postal application in Australia and contains problem instances of up to 200 nodes. The AP data set has been used for both the capacitated and uncapacitated SAHLP.

The performance of the GA on both sets of the benchmark problems is very good. It finds optimal or known-best solutions for most of the SAHLP problems. For some of the large-sized capacitated SAHLP problems, it finds improved solution. Moreover, it outperforms the current GA [13] for the capacitated SAHLP both in terms of solution quality and number of problems solved to optimality or current best values in SAHLP literature. For larger AP problems, it outperforms the simulated annealing (SA) and random descent heuristic (RDH) based hybrid approach to CSAHLP. For the uncapacitated SAHLP, the performance of the proposed GA is better than that of GA [3] and GATS [4] on CAB problems and comparable with that of GA [3] and SATLUHLP [11] on AP problems.

Overall, the proposed GA yields high-quality solutions to most of the benchmark problems. The computational time of the GA is also satisfactory on the given benchmark and platform. It solves most of the small-sized benchmark (AP and CAB) problems i.e., problems with 50 or less nodes, in less than 100 seconds. Its computational time for large-sized problems i.e., problems with more than 50 nodes, is also within reasonable limits not exceeding few hundred seconds.

The rest of the layout of this work is as follows.

Chapter 2 provides the background to the problem. Section 2.2 describes the problem. Section 2.3 covers present work on the Single Allocation Hub Location Problem (SAHLP) and section 2.3. gives a brief overview of the Genetic Algorithm (GA). Chapter 3 covers the proposed GA approach. Solution encoding schemes for SAHLP are discussed in section 3.1. Crossovers are explained in section 3.2. Section 3.5 presents the constraint handling technique employed in the proposed GA approach. Experimental setup and computational results are presented in Chapters 4 and 5. Chapter 4 presents computational results for the uncapacitated SAHLP and chapter 5 for the capacitated SAHLP. Sections 4.1 describes experimental setup and parameters, section 4.2 gives a description of the AP data set, and section 4.3 discusses computational results and performance of the three GA implementations (i.e., GA-1, GA-2, and GA-3). The same structure has been retained for chapter 5. The CAB data set is described in detail in section 5.2 and computational results of three implementations of the GA for the CAB data have been given in section 5.4. Chapter 6 gives statistical analysis of the GA performance and Chapter 7 derives conclusions and discusses the scope for future work.

Chapter 2

Background

This chapter provides an overview of the Single Allocation Hub Location Problem(SAHLP). A summary of the current metaheuristics for the SAHLP is also given. Lastly, the basics of genetic algorithms are provided.

Hub-and-spoke networks are distribution systems in which some nodes, called *hubs*, serve as switching, sorting, or transhipment centers for the flow of commodities while the remaining nodes, called *spokes*, function as the origin or destination points. The networks seek to reduce the overall transportation cost by consolidating traffic flows from different origins but to same destination at a hub-point and shipping them to the destination. An example of a hub-and-spoke network is shown in Figure 2.1. In this figure, nodes i, j, k, and l are hubs whereas the rest of the nodes are spokes. As the figure shows, in a hub-spoke network, the hub subnet is a complete graph whereas spokes are connected only to hubs. Thus all the internodal flow or communication in the network occurs through hubs.

The total cost of the commodity flow in a hub-spoke network is determined by the cost associated with location of hubs in the network and the cost of allocating spokes to hubs. Locating hubs and allocating spokes to hubs in a hub-spoke network is an NP-hard combinatorial optimization problem called Hub Location Problem (HLP). The objective of the hub location problem is to minimize the cost of commodity flow or transportation through the network. Hub location problem has two main types. In the Single Allocation Hub Location Problem (as shown in Figure 2.2) , a node can be assigned to only a single hub whereas in Multiple Allocation Hub Location Problems (Figure 2.1), it can be assigned to multiple hubs. This work focuses on the Single Allocation Hub Location Problem (SAHLP). A formal description of the SAHLP is provided in the following sections.

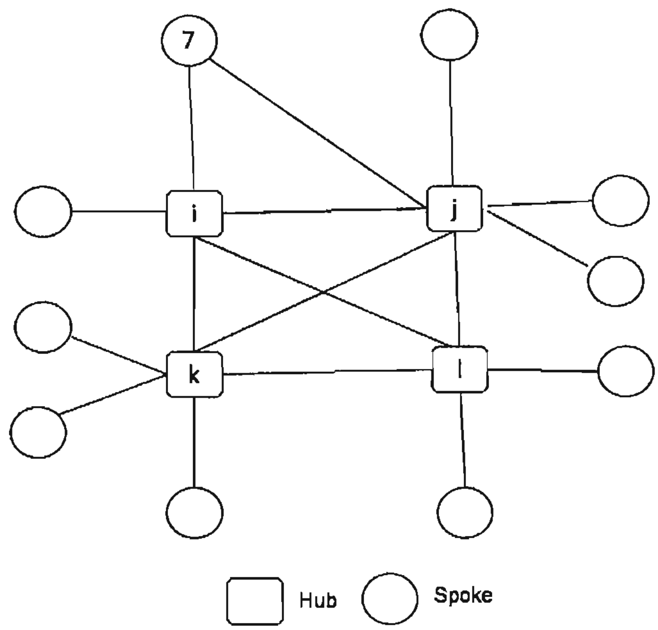


Figure 2.1: A hub-spoke network

2.1 The Single Allocation Hub Location Problem(SAHLP)

The single allocation hub location problem is a special type of hub location problem in which a spoke can be assigned to only a single hub. Moreover, the number of hubs is a decision variable in SAHLP and a fixed cost for establishing a hub is also included in the overall transportation cost. A single allocation hub-and-spoke network is shown in the Figure 2.2. The problem involves the following decisions.

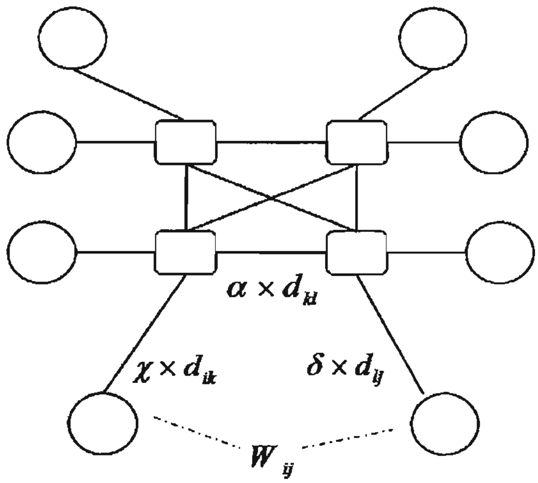


Figure 2.2: A single allocation hub-spoke network

- Determining the number of hubs to be used.
- Location of hubs i.e., where in the network should the hubs be located?
- Allocation of spokes to hubs i.e., how are spokes to be assigned to hubs?

The objective in the SAHLP is to minimize the cost of establishing hubs and cost of transportation. This is subject to the constraints that a spoke must be assigned to only a single hub, flows must be routed through hubs (at least one and at most two), and hub capacities must not be exceeded. The transportation cost in single allocation hub location problem has the following three components.

- The *Collection cost*, χ, is the cost incurred on flow from a given spoke to a hub i.e., cost of spoke-to-hub flow.

- The *Transfer cost*, α, represents the cost of the flow between hubs i.e., cost of hub-to-hub flow.

- The *Distribution cost*, δ, denotes the cost of the flow from a hub to a spoke i.e., cost of hub-to-spoke flow.

All the cost types are per unit distance of flow volume between nodes. For example, assume that in Figure 2.2, W_{ij} volume of a commodity is sent by node i to node j. W_{ij} is first transported from node i to hub k, then from hub k to hub l, and finally from hub l to the destination node j. The net transportation cost C_{ijkl} is,

$$C_{ijkl} = W_{ij} \ (\chi d_{ik} + \alpha d_{kl} + \delta d_{lj})$$

Where d_{ik} is the distance between node i and hub k, d_{kl} is distance between hubs k and l, and d_{lj} is the distance between hub l and node j. In order to find the transportation cost of the entire network, C_{ijkl} is calculated for all the node pairs in the network. The cost of establishing the required hubs is also included in the total cost.

The SAHLP has two varieties; (1) the uncapacitated Single Allocation Hub Location Problem (USAHLP) and (2) the capacitated Single Allocation Hub Location Problem (CSAHLP). In the uncapacitated SAHLP, hubs can handle unlimited flow from other nodes whereas in the capacitated SAHLP, hubs have capacity limits imposed on them. This work investigates both the capacitated and uncapacitated variants of the Single Allocation Hub Location Problem. It uses the CSAHLP-C formulation for the Capacitated Single Allocation Hub location Problem(CSAHLP) proposed by A. T. Ernst and M. Krishanmoorthy [1]. CSAHLP-C is a mixed integer formulation and can be found in Ernst *et al.* [1]. The formal description is given below.

Minimize $\sum_{i\in N_N}\sum_{k\in N_N}\sum_{l\in N_N}\sum_{j\in N_N} W_{ij}(\chi d_{ik} + \alpha d_{kl} + \delta d_{lj})X_{ijkl} + \sum_{k\in N} F_k Z_{kk}$

Subject to:

$$\sum_{k\in N_N}\sum_{l\in N_N} X_{ijkl} = 1, \qquad \forall i,j \in N_N, \quad (1)$$

$$Z_{ik} \le Z_{kk}, \qquad \forall i,k \in N_N, \quad (2)$$

$$\sum_{j\in N_N}\sum_{l\in N_N}(W_{ij}X_{ijkl} + W_{ji}X_{jilk}) = (O_i + D_i)Z_{ik}, \qquad \forall i,k \in N_N, \quad (3)$$

$$\sum_{i\in N_N} O_i Z_{ik} \le \Gamma_k Z_{kk}, \qquad \forall i,k \in N_N, \quad (4)$$

$$Z_{ik} \in \{0,1\}, \qquad \forall i,k \in N_N, \quad (5)$$

$$0 \le X_{ijkl} \le 1, \qquad \forall i,j,k,l \in N_N, \quad (6)$$

Where:
$O_i = \sum_{j\in N_N} W_{ij}$
$D_i = \sum_{j\in N_N} W_{ji}$
N is the number of nodes.
$N_N = \{0, 1, 2, ..., N-1\}$
W_{ij} is the amount of flow between the origin i and destination j.
χ is the *collection* cost (from origin spoke to hub).
α is the *transfer* cost (between hubs).
δ is the *distribution* cost (from hub to destination spoke).
d_{ik} represents the distance between nodes i and hub k.
d_{kl} is the distance between hubs k and l.
d_{lj} is the distance between hub l and node j.
X_{ijkl} is the decision variable that represents the fraction of traffic between origin node i to destination node j through hubs k and l.
F_i is the cost of establishing node i as hub.
Γ_i is the capacity of hub i.
Z_{ij} is 1 if node i is assigned to hub j, otherwise it is 0.
Z_{kk} is 1 if node k is also a hub, otherwise it is 0.

Constraint (1) ensures that all the traffic between an origin-destination pair has been routed via the hub sub-network. Constraint (2) prevents non-hub nodes from being allocated to other non-hub nodes while Constraints (3) and (4) restrict the commodity flow through each hub (i.e., each hub has a limited capacity). For some hub-spoke networks e.g., a mail delivery system, the problem may not be symmetric i.e., $W_{ij} \neq W_{ji}$. Additionally, it may be the case that $W_{ii} > 0$ so that a node may route commodities to itself. In this work, both symmetric and non-symmetric flows are employed. The above formulation except for the capacity constraints (3) and (4) also applies to the uncapacitated SAHLP.

2.2 Previous Work

The Single Allocation Hub Location Problem is NP-hard combinatorial optimization problem [10]. Due to this complexity, solving it with exact methods is computationally intractable especially when large problem instances are involved. Therefore, in recent years, meta-heuristics such as Genetic Algorithms (GAs) [3][4][12][13], Tabu Search(TS) [11], and Ant Colony Optimization (ACO)[2] algorithms have been proposed for the SAHLP. A brief overview of the solution approaches applied to both the capacitated and uncapacitated versions of the SAHLP is given next.

2.2.1 Uncapacitated SAHLP (USAHLP)

Abdinour-Helm [4] proposed a hybrid approach based on GA and Tabu Search to solve the USAHLP. The GA was used to determine the number and location of hubs and the Tabu Search(TS), to assign spokes to hubs. They reported an improvement over their earlier GA-approach that used distance-based assignment of spokes to hubs. However, their stand-alone GA results are not available. Topcuoglo *et al.* [3] developed a GA-based approach to the USAHLP. They found improved solutions to some Civil Aviation Board (CAB) problems. They also used Australian Post (AP) data in their experiments that had not been previously used in any study on USAHLP. Another GA-based study on the USAHLP cited by Kara *et al.* [10] is a hybrid approach by Cunha and Silva [12] that employed GA and Simulated Annealing .

The non-GA heuristics applied to the USAHLP include two hybrid approaches by Chen *et al.* [11] and Silva *et. al.* [26]. Chen *et al.* [11] combined SA with Tabu List(TL) to solve USAHLP. This approach involves applying Simulated Annealing to determine an upper-bound for the number of hubs and then using restricted single location exchange procedure to locate the hubs. Non-hub nodes are first allocated to nearest hubs followed by an improvement procedure for allocation that iteratively re-allocates nodes with less flow to other hubs until no improvement is possible.

2.2.2 Capacitated SAHLP (CSAHLP)

To the best of the author's knowledge, the only GA-method for the CSAHLP is by Stanimirovic *et al.* [13]. Stanimirovic e al. [13] improves solutions for some of the larger problems for CSAHLP.

Ernst *et al.* [1] proposed a mixed integer formulation for the CSAHLP and developed two heuristic algorithms for the problem based on simulated annealing and random descent. They used the upper bound obtained with SA-RDH to develop an LP-based branch and bound solution method for CSAHLP). Ernst *et al.* [1] also introduced the AP (Australian Post) benchmark data for the Capacitated Single Allocation Hub Location Problem(CSAHLP), which has since been used by various research works including the one presented here. Randall *et al.* [2] applied Ant Colony Optimization (ACO) algorithm to CSAHLP. In their work, four variants of ACO algorithm each based on a different construction modeling choice and combined with multiple neighbourhood search to solve CSAHLP were developed.

As the above summary reveals, although genetic algorithms have been applied to the SAHLP, a narrow range of GA based approaches have been used to tackle the SAHLP. Moreover, only one GA has been proposed to CSAHLP. There is thus room for further GA contribution to the single allocation hub location problem. This work proposes a GA approach to SAHLP based on new solution encodings and crossovers as presented in chapter 3. A brief overview of the fundamentals of genetic algorithms is given next.

2.3 Genetic Algorithms

Genetic Algorithms (GAs), introduced by James Holland in his seminal work "Adaptation in natural and artificial systems" in 1975, are a family of population based stochastic computational methods inspired by biological evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators and selection process to these data structures so as to evolve better solutions and preserve critical information [29].

GAs have successfully been applied to a broad range of problems for example bin-packing [30], job scheduling [31], vehicle routing [23], location-allocation [32], etc. problems. Similar to other metaheuristics, while they are not guaranteed to yield an optimal solution to a given problem, they can provide good approximations within acceptable time as opposed to using an exact method for an optimal solution, which would be computationally intractable for larger problem instances.

GAs operate on a data structure called *chromosomes*. A chromosome is a direct or indirect representation of a solution-entity in the real-world. The configuration of attributes that gives rise to a solution-entity is called the *phenotype*. In direct representation, the phenotypic information is directly used in the chromosome i.e., each attribute in the phenotype is encoded by a gene. Indirect representation involves more complex genotype-to-phenotype mapping but is flexible and can be easily processed by recombination and mutation operators. Different encoding schemes are used to encode phenotypic information in the genotype. These include binary representation, integer representation, tree representation, etc. [22] The choice of an encoding scheme for a given problem depends on the solution structure of the problem [22].

A GA typically starts with an initial population of chromosomes (representing solutions), which is usually randomly chosen. The quality of the solutions is iteratively improved by evaluating the individual fitness of the population and stochastically choosing parent solutions from the population to reproduce children solutions through a recombination operation. An optional mutation operation is also applied to maintain diversity in the population. A description of the GA is given in Algorithm 1. Explanation of its different components is given in the subsequent sections.

Algorithm 1 Genetic Algorithm

generate initial population
repeat

Evaluate the individual fitnesses of the population.
Select pairs of individuals from the
population to reproduce.
generate a new pool of the population through
crossover and mutation.

until *terminating – condition*

2.3.1 Generation of initial population

Initially, a population of individual solutions is generated to form an initial population. Traditionally, the population is generated randomly. Occasionally, the solutions may be "seeded" with solutions biased towards areas where "optimal" solutions are likely to be found, but this is not a standard requirement by all GAs.

2.3.2 Fitness Evaluation

Each generation of population undergoes fitness evaluation. The fitness of an individual is evaluated using some function evaluation. Values of the fitness function indicate the cost of the solutions of the population in a generation and provide a basis for identifying fitter solutions in the subsequent selection process.

2.3.3 Selection

During each successive generation, a proportion of the existing population is selected to breed a new generation. A number of selection methods have been developed to identify individuals for reproduction. Some selection methods rate the fitness of each solution and preferentially select the best solutions for cross-breeding [28][33]. Other methods rate only a random sample of the population and choose the better individuals [28][33]. Sometimes, solutions are stochastically selected from the sample so that less fit solutions also have the chances of selection. This preserves the diversity of the population helps the algorithm avoid premature conversion on poor solutions. Popular and well-studied selection methods include roulette wheel selection [33][35] and tournament selection [33][35].

2.3.4 Reproduction

The next phase is to produce a new population of solutions from the individuals selected for cross-breeding. For this purpose a pair of individuals is selected and, probabilistically, subjected to recombination in the form of crossover and mutation operators. During the recombination operation, the individuals in the selected pair mate by exchanging genetic materials and in the mutation operation, a random gene of the solution is changed. The resulting offspring are added to the next-generation population pool.

A variety of recombination operators have been explored in literature. Some recombination operators like *1-point*[27][34][35], *2-point* [27][34][35], and *uniform-order* [27][34][35], PMX[35][38], and Cycle [35][38] crossovers are considered general and applicable to a broad range of problems. Other crossovers are designed for specific problem-families taking into account their solution structures and may incorporate domain-specific knowledge to build better solutions quickly [39][40].

Like crossover operators, there are also different kinds of mutation operations suited to specific problems available [27]. For ordering problems, for examples TSP, different mutation operations like *inversion* [33][35] and *exchange* [33][35] mutations are used. Similar to recombination, choice of mutation operation for a problem depends on the nature of the problem.

2.3.5 Termination Criteria

Two strategies are employed to terminate the algorithm. More often, the GA is terminated after it executes for a specified number of generation span. Sometimes, it is terminated when no progress is made in the quality of solutions over a given number of successive generations.

Chapter 3

Genetic Algorithm for the Single Allocation Hub Location Problem

In this chapter, a GA approach to the Single Allocation Hub Location Problem is proposed. Some new crossover and representation techniques employed in the proposed GA are presented. Finally, an efficient technique to handle the capacity constraint in the Capacitated Single Allocation Hub Location Problem is provided.

The GA-approach to the SAHLP proposed in this thesis is based on three new crossovers i.e. the *Double-Cluster Exchange Crossover (DCEC)*, the *Multi-Cluster Exchange Crossover (MCEC)*, and the *Best-Cost Routing Crossovers (BCRC)*. It uses two new representations schemes i.e. *List-based* and *Set-based* representations for the SAHLP. The $k-4$ Tournament Selection method [28] is used in the GA. Moreover, to handle the capacity constraint in the capacitated version of the SAHLP, the GA employs the repair technique.

The GA methodology for the SAHLP is given by Algorithm 2. The GA starts by randomly generating an initial population of chromosomes representing potential solutions. The chromosomes are then subjected to an evolutionary process until a minimal cost hub-spoke network is evolved or the termination condition is met. The evolutionary process has the same structure as in ordinary GA using crossover and selection operations on chromosomes. Three problem-specific crossovers i.e., MCEC, DCEC, or BCRC based on cluster exchange between the mating parents are used in the GA to produce offspring. The GA incorporates three mutation operations i.e., *shift*, *exchange*, and *replace-hub* - chosen probabilistically - to maintain population diversity. It employees tournament selection with elitism [29] to perform fitness-based selection of individuals for evolutionary reproduction. Infeasible solutions are repaired in case of the capacitated SAHLP. Each of these GA components is described in the following sections.

Algorithm 2 GA for the SAHLP

1. *Generate an* **initial population**
2. **Evaluate** *the fitness $F(x)$ of each chromosome x of the population and calculate the average fitness;*
3. *Create a new population by repeating the following steps until the new population is complete;*
   - **Selection** *Select two chromosomes from the population using tournament selection;*
   - **Reproduction** *Apply MCEC, DCEC, or BCRC crossover probabilistically to parents to form new offspring. If crossover is not performed, offspring is an exact copy of parents;*
   - **Mutation** *With mutation probabilities, apply shift, swap, and replace-hub mutations;*
   - **Acceptance** *Place the offspring in the population replacing the parents;*
   - **Elitism** *Replace 4 randomly-chosen individuals of the population with 4 best individuals from the parent population;*
4. *Update the old population with the newly generated population;*
5. *If the present number of generations is reached, stop, return the average fitness, and the fitness of the best chromosome in the current population;*
6. *Else go to step 2;*

3.1 Solution Encoding and Initial Population Creation

A good solution representation is of critical importance to GA performance. It captures and helps propagate the basic building blocks of the solution for the target problem[22][41]. The proposed GA employees two representation schemes i.e., *List-based Representation* and *Set-based Representation,* to encode the solution structure of the SAHLP. Both these schemes use integers to represent nodes in a network e.g. an $n{-}nodes$ network is represented by integers in the range $0 \ldots n-1$. To better capture the location-allocation information in the representations, a hub-spoke network is considered to be a combination of hub-spoke clusters. An example is given in Figure 3.1. In this network, there are 12 nodes organized into one hub-hub subnet comprising nodes 1, 12, and 8, and 3 hub-spoke subnets (here called clusters). These clusters are $C1 = \{\underline{1}, 6, 3, 4, 11\}$, $C2 = \{\underline{12}, 5, 10, 2\}$, and $C3 = \{\underline{8}, 7, 9, 0\}$ with nodes 1, 12, and 8 as hubs and the remaining nodes as the spokes. Standard crossovers don't apply to these representations. The representations are described next.

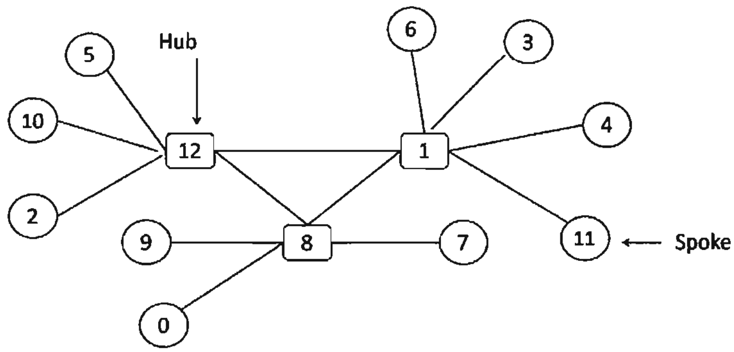


Figure 3.1: A hub-spoke network

3.1.1 Set-based Representation

In the set-based Representation, a chromosome is a collection of sets of numbers in which each number represents a node. A set maps a cluster in the network. The first number in the set represents a hub and the remaining numbers represent the spokes allocated to the hub. An example of this encoding scheme is given in Figure 3.2.

The network in the Figure 3.2 (a) has three clusters i.e., $C_1 = \{\mathbf{12}, 5, 10, 2\}$, $C_3 = \{\mathbf{1}, 6, 3, 4, 11\}$, and $C_2 = \{\mathbf{8}, 9, 0, 7\}$. As typed in the boldface, 12, 8, and 1 are hubs in the clusters. Thus, there are three sets, S_1, S_2, and S_3, representing clusters C_1, C_2, and C_3, respectively as shown in Figure 3.2 (b). The first number in each set is a hub and the remaining nodes are spokes associated with the hub. For example, in the first set, node 12 is the hub and node 5, 10, and 2 are the spoke assigned to hub 12.

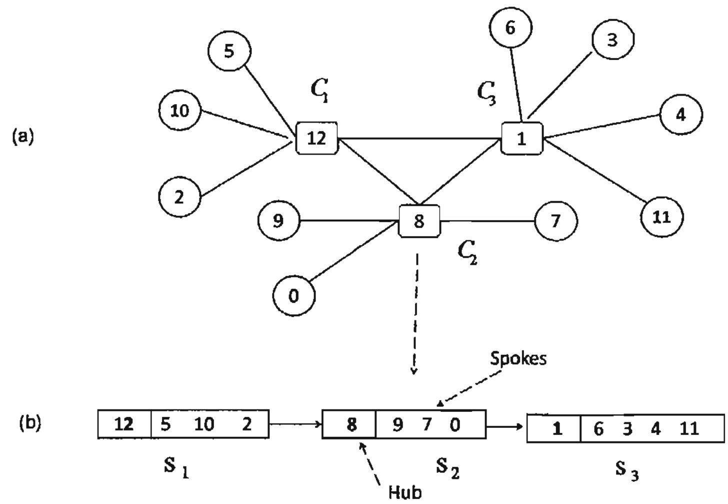


Figure 3.2: Set-based chromosome: (b) represents chromosome with corresponding network shown in (a)

3.1.2 List-based Representation

In the list-based encoding, a solution is represented by a list. The list has n hub entries where n is the number of nodes in the network. The entries are also implicitly indexed by numbers from 0 to $n - 1$ that represent spokes. Thus a hub entry in the list is indexed by one of the spokes assigned to the hub. This representation scheme is illustrated in the Figure 3.3.

The network in the Figure 3.3 (a) has 13 nodes including pre-designated hubs i.e. 1, 8, and 12. Thus its list representation contains 13 entries as shown in Figure 3.3 (b) . Every list entry is a hub i.e., either 1, 8, or 12. The hub entries are numbered 0 to 12, such that 0 serves as an index to the first value in the list, 1 to the second value, 2 to the third value, and so on. The hub entry at position 0 of the list is 8, which means spoke 0 is assigned to hub 8. Similarly, hub at position 1 is 1 meaning spoke 1 is assigned to hub 1 and at position 2 is 12 indicating that spoke 2 is allocated to hub 12, etc. In SAHLP formulation, a hub is considered to be assigned to itself. This is indicated by storing values 1, 8, and 12 at positions 1, 8, and 12 of the list.

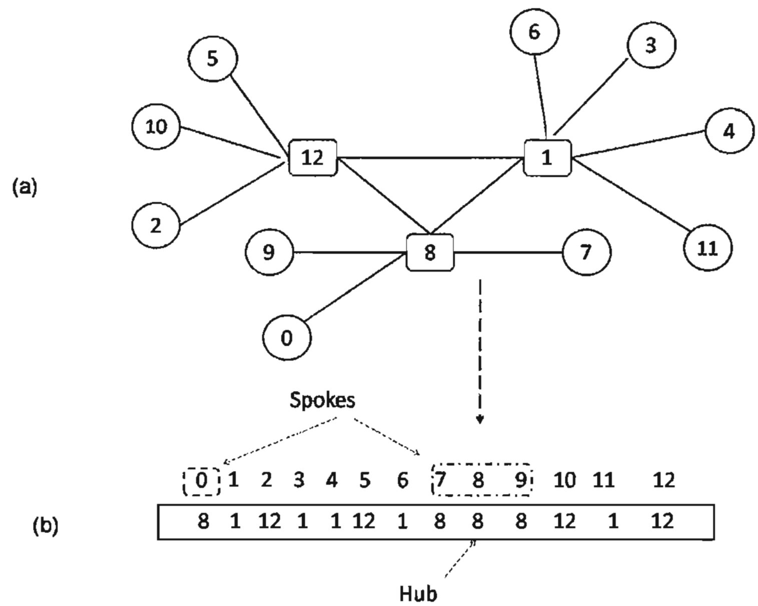


Figure 3.3: List-based chromosome

3.1.3 Initial Population Creation

The creation of an individual is performed in three steps. In the first step, the *number of hubs* is determined randomly. The maximum number of hubs in this work can be half the number of nodes in the network and the minimum 2. In the second stage, i.e., *Location Step*, m hubs are randomly chosen from n nodes (n is the number of nodes in the network). In this way, any node from 1 to n has the chance to become a hub. Lastly, i.e., in the *Allocation Step*, the remaining $n - m$ nodes are allocated to the selected hubs using the distance-based assignment rule i.e., a given node is assigned to a hub that has the shortest (Euclidean) distance from the node. In the final *repair step*, the solution is repaired if the hub-capacity constraint is violated by an assignment. Repair step is required only for the capacitated SAHLP. Section 3.5 gives a detailed description of the repair procedure.

The above process is applied iteratively to create the entire population. It is given in Algorithm 3.2 and illustrated in Figure 3.4 for the *list representation* with an arbitrarily chosen network of 12 nodes. In the first step, the number of hubs i.e. $m = 3$ is determined using the formula given above. In the *location step*, 3 random nodes i.e. 1, 8, and 12 are selected as hubs and inserted in the list at positions 1, 8, and 12.. In the *allocation step*, the remaining nodes are assigned to 1, 8, and 12 e.g. the nearest hub to spoke 5 is 12, so 12 is inserted at position 5 in the list. Likewise, 3 is assigned to its nearest hub 1 (1 is entered in the list at position 3). Other nodes are assigned in the same way. In the last step i.e., the *repair step*, node 7 is shifted from hub 1 to hub 8 to handle capacity overflow for hub 1.

The process for the set representation is given in Figure 3.5. In the *location* step, three nodes i.e., 1, 8, and 12 are randomly selected as hubs and inserted in three sets, one set for each hub. In the *allocation* step, remaining nodes are inserted in the sets according to distance from the hubs e.g. node 5 is placed in the first set as in the figure because its nearest hub is 12. In the *repair step* that is required in the capacitated SAHLP, a solution is repaired if a hub overflow occurs due to the assignment of nodes to hubs. In the figure, this is depicted through shifting node 4 from overflow hub 1 to hub 12. A set with the hub and assigned nodes, thus, maps a cluster in the hub-spoke network.

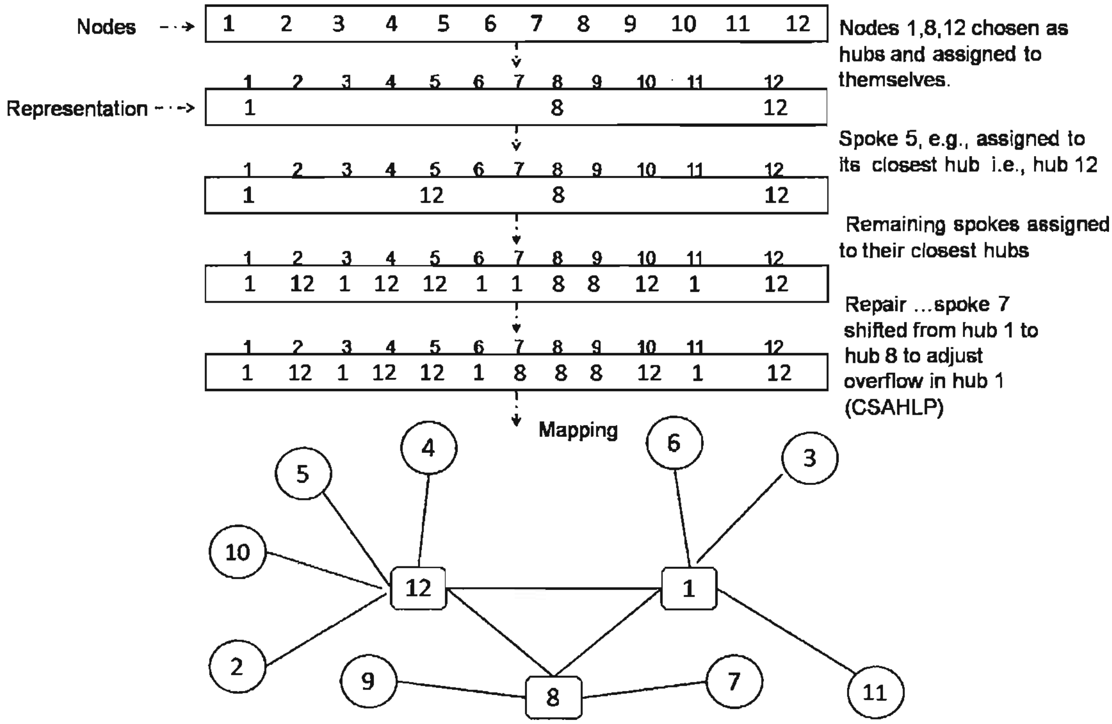


Figure 3.4: Individual creation in initial population generation: list representation

Algorithm 3 Initial population generation

1. **Select** *at least* 2 *and at most* $n \div 2$ *hubs from the total* n *nodes in the network*;
2. **Locate** *selected number of nodes to serve as hubs*;
3. **Allocate** *the remaining nodes to the selected hubs using the distance assignment rule i.e., assign nodes to their closest hubs*;
4. *If the problem type is capacitated SAHLP, invoke the* **Repair-Module** *to repair the individual if it is infeasible*;
5. *If required number of individuals have been created exit otherwise go to step 1*;

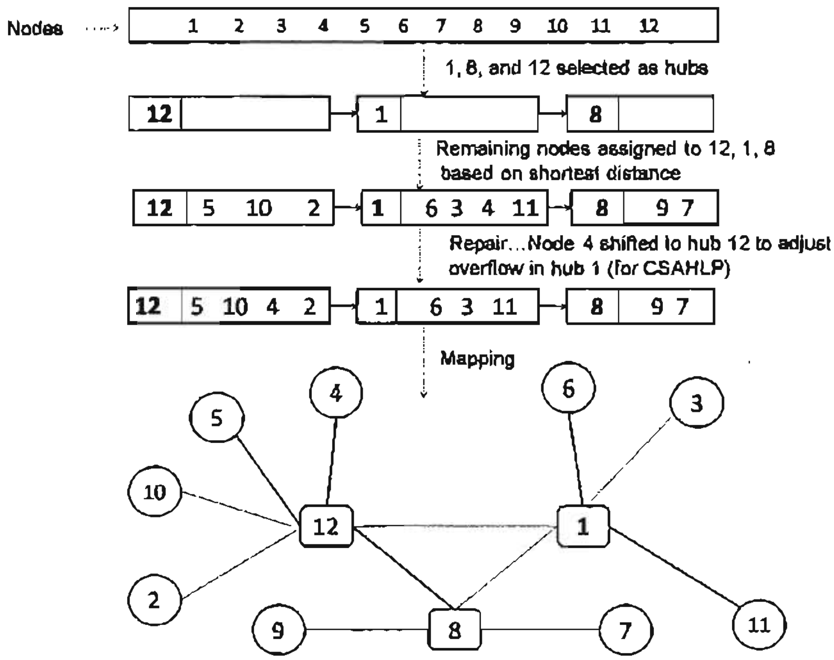


Figure 3.5: Individual creation in initial population generation: set representation

3.2 Evaluation of Chromosome Fitness

The proposed GA uses the objective function in CSAHLP-H formulation [1] given below as the fitness $f(x)$ of a chromosome.

$$F(x) = \sum_{i \in N} \sum_{l \in N} \sum_{k \in N} \sum_{j \in N} W_{ij}(\chi d_{ik} + \alpha d_{kl} + \delta d_{lj}) X_{ijkl} + \sum_{k \in N} F_k Z_{kk}$$

In the above function, the first term represents the cost incurred on the internodal flow and the second term the cost of establishing the selected nodes as hubs. The function as a whole represents the total transportation cost of the network.

To evaluate the fitness of a chromosome, its representation is first mapped into an allocation array. For list-based representation, this mapping is not necessary because the representation is already in the required form. For the set-based representation, the step is shown in the Figure 3.6. Figure 3.6 (a) is the set-based representation of a chromosome and Figure 3.6 (b) is its allocation array. After mapping into an allocation array, the fitness of a chromosome is evaluated according to Algorithm 3.3

In the algorithm, n is the number of nodes in the network and f is the value of the objective function. The symbol h_i represents an element of the allocation array in Figure 3.6. For example, element h_0 of allocation array in Figure 3.6 has value 0 and h_{10} has value 12. The notation w_{ij} denotes the flow volume between nodes i and j whose value is specified by the flow matrix below.

$$\begin{vmatrix} w_{11} & w_{12} & \cdots\cdot & w_{1n} \\ w_{21} & w_{21} & \cdots\cdot & w_{2n} \\ & & & \\ & & & \\ w_{n1} & w_{n2} & \cdots\cdot & w_{nn} \end{vmatrix}$$

The expression d_{ij} is the (Euclidean) distance between nodes i and j. Its value is given by the distance matrix in figure 3.6.

Furthermore, $cCost$, $tCost$, and $dCost$ are collection(χ), transfer(α), and distribution (δ) costs respectively and $hCost[k]$ is the cost of hub k. Distance and volume matrices, transportation cots (i.e. $cCost$, $tCost$, and $dCost$), and hub costs (i.e. $hCost[k]$) are specified by the benchmark to test the algorithm.

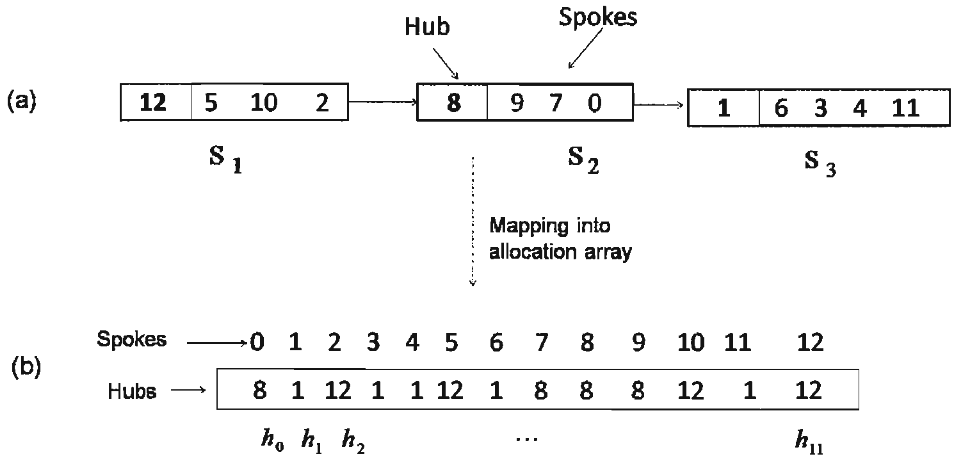


Figure 3.6: Mapping from set-based representation to allocation array

$$\begin{vmatrix} d_{11} & d_{12} & .... & d_{1n} \\ d_{21} & d_{21} & .... & d_{2n} \\ & & & \\ & & & \\ d_{n1} & d_{n2} & .... & d_{nn} \end{vmatrix}$$

Algorithm 4 Evaluation of chromosome fitness

$f \leftarrow 0.0$

{Calculation transportation cost of the solution}

for $i = 0 \;\; to \;\; n-1$ **do**

  for $j = 0 \;\; to \;\; n-1$ **do**

    $f \leftarrow f + w_{ij} * (cCost * d_{ih_i} + tCost * d_{h_ih_j} + dCost * d_{h_jj})$

  end for

end for

{Add hub cost of the solution}
{A hub is added to the set $\wp$ after its cost has been processed}

for $k = 0 \;\; to \;\; n-1$ **do**

  if h_k not in $\wp$ **then**
    $f \leftarrow f + hCost_k$
    $\wp \leftarrow h_k$
  end if

end for

3.3 Reproduction

Selection of individuals from the population for mating forms an important component of the evolutionary process in a genetic algorithm. There are many selection schemes available. Choice of an appropriate selection method is a crucial step in the application of genetic algorithms to problems. The proposed GA employes tournament selection with a tournament size of 4 for selecting the mating individuals. Tournament sizes of 2 was also tested but proved to be inefficient. The tournament size of 4 on the other hand gave better performance in terms of convergece on the SAHLP.

3.4 Crossovers

The performance of a crossover for a problem depends on its efficient propagation of the building blocks of the solution of the problem. In the Single Allocation Hub Location Problem, combinations of hubs and hub-spoke assignment patterns constitute the building blocks of the solution. Further, the fitness contribution of a cluster in terms of minimizing the objective function depends on the distance and flow between spokes and the hub in the cluster.

Based on the above observations, two problem-specific crossovers were designed for the SAHLP that process clusters instead of individual nodes. A cluster in this context is a hub and its allocated spokes (Figure 3.1, section 3.1). In these crossovers, one or more clusters are exchanged between the mating parents. Thus a cluster forms the unit of gene-exchange in such crossovers. Two crossovers i.e., Multi-Cluster Exchange Crossover (MCEC) that exchanges multiple clusters between the mating parents and Double-Cluster Exchange Crossover (DCEC) that exchanges two clusters between the mating parents were designed during the course of the proposed GA development. Crossovers based on partial and single cluster exchange were also implemented but proved inefficient for larger SAHLP problems. The third crossover is an adaptation of the Best Cost Routing Crossover (BCRC) employed by Ombuki *et. al.* [23] [42] for Vehicle Routing Problems with time windows(VRPTW). These crossovers are described in detail in the following sections.

3.4.1 Multi-Cluster Exchange Crossover(MCEC)

In the Multi-Cluster Exchange Crossover, children solutions are produced by swapping one or more randomly selected clusters between the mating parents. The swapping process is followed by a re-adjustment process in which infeasible solutions are corrected. If a hub in a cluster from one parent (i.e., the source parent) is also a hub in a cluster of the other parent (i.e. the destination parent), then both clusters are merged. Duplicate or missing nodes in a child solution resulting from this process are re-assigned based on distance i.e., a node is assigned to the nearest hub. Stand-alone hubs i.e., hubs without spokes, resulting from the recombination operation are assigned to other hubs as spokes. For illustration, consider two parent solutions P_1 and P_2 with their respective networks in Figure 3.7 selected for cross-breeding. Multi-Cluster Exchange Crossover (MCEC) is applied to P_1 and P_2 to produce two children solutions Ch_1 and Ch_2 as described below.

1. First is the *selection* step in which two clusters, $C_1P_1 = \{\mathbf{4}, 5, 8, 9\}$ and $C_3P_1 = \{\mathbf{7}, 2\}$, are randomly selected from parent P_1 and one cluster, $C_2P_2 = \{\mathbf{2}, 7, 6, 5\}$ from parent P_2.

2. Next is the *swapping* step where clusters C_1P_1 and C_3P_1 are removed from P_1 and added to P_2. Likewise, C_2P_2 is removed from P_2 and added to P_1. In this way, offspring Ch_1 and Ch_2 are produced as shown in the Figure 3.8.

3. Then in the *merger* step, $C_3 = \{\mathbf{4}, 3\}$ and $C_2 = \{\mathbf{4}, 5, 8, 9\}$ in Ch_2 are merged into a single cluster because they have the same hub (i.e., hub 4). Result is the children solutions as given in Figure 3.9.

4. Next, in the *re-adjustment step*, duplicate nodes in an offspring are re-allocated and missing nodes re-inserted. This is necessary because duplicate nodes violate the single-assignment constraint of the network and missing nodes violate the network integrity. Ch_2 has duplicate nodes 8, 9. They are detached from their present hubs and re-assigned to other hubs in Ch_2 based on distance. Missing node 6 from the child, Ch_2, is re-inserted into it in the same way. This step is repeated for Ch_1. The result is shown in Figure 3.10.

5. Final step is the *removal and re-assignment of stand-alone hubs*. This is unnecessary in the present example because there is no such hub either in Ch_1 or Ch_2.

The minimum number of clusters that can be selected from a parent solution to exchange with the other parent can be 1 to $k - 1$ where k is the number of clusters in the solution. If a parent solution has 2 or less clusters, then 1 cluster is

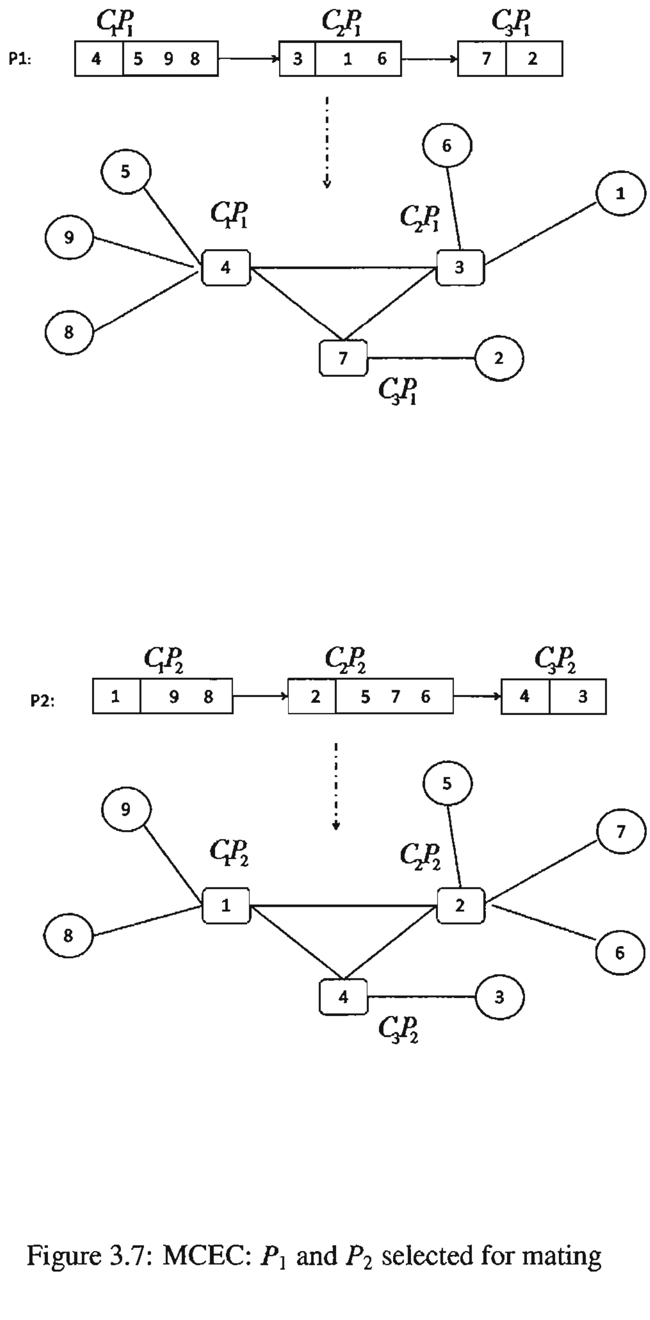
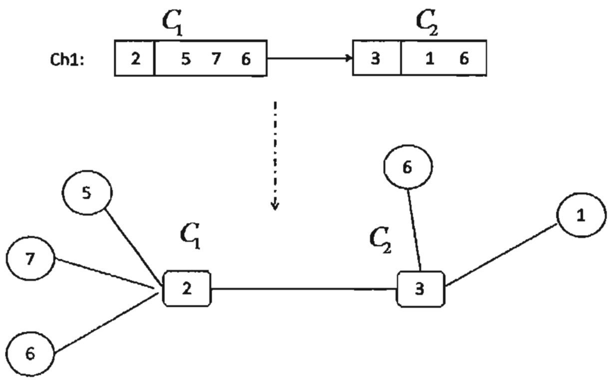


Figure 3.7: MCEC: P_1 and P_2 selected for mating



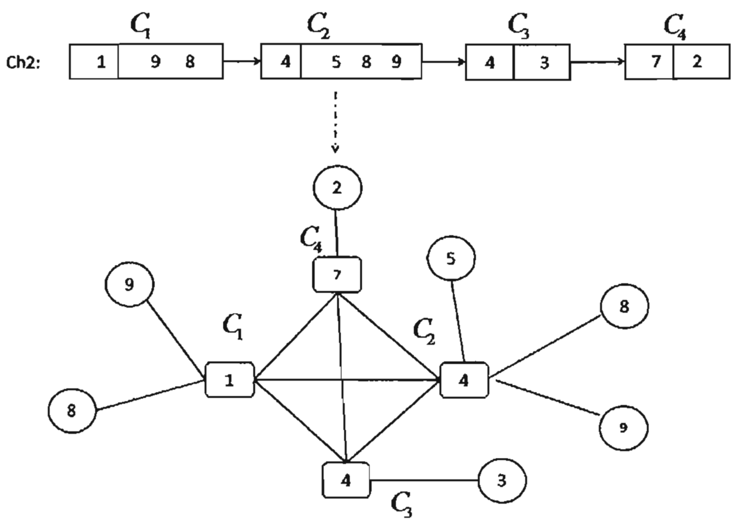


Figure 3.8: MCEC: Children solutions after the *swapping* step

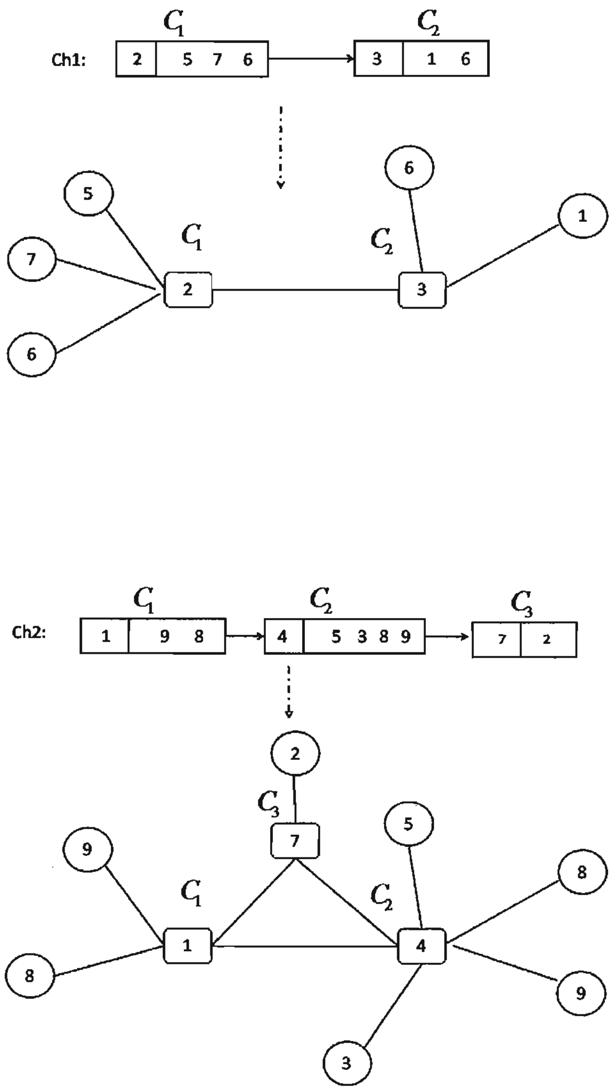


Figure 3.9: MCEC: Children solutions after the *merger* step

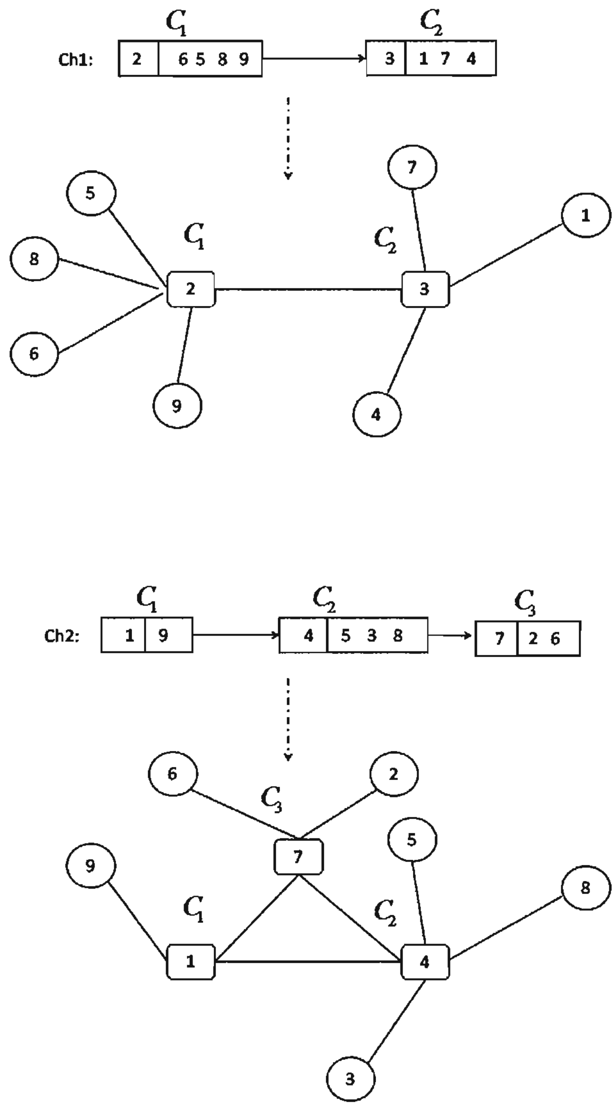


Figure 3.10: MCEC: Children solutions after the *re-adjustment* step

selected to swap with the other parent. Moreover, while re-inserting a node into an offspring, a hub that is the closest to the node and has the required capacity is selected. If no such hub is found, then the node is assigned to the closest hub regardless of the violation of capacity.

3.4.2 Double-Cluster Exchange Crossover(DCEC)

In Double-Cluster Exchange Crossover (DCEC), two random clusters are iteratively selected from one parent solution and shifted to the other. The same operation is repeated for the other parent. Duplicate nodes in the offspring resulting from the recombination operation are detached from their present hubs and re-assigned to other hubs of the same offspring according to distance i.e., a node is assigned to the closest hub. Likewise, nodes lost by an offspring due to the swap operation are re-inserted in it based on distance. Stand-alone hubs i.e., hubs without spokes, in a child solution are demoted to spokes and re-assigned to other hubs.

The process is illustrated In Figure 3.11. There are two parent solutions, $P_1 = \{\{3,1,2\},\{4,5,8,9\},\{7,6\}\}$ with hubs 3, 4, 7 and $P_2 = \{\{1,9,8\},\{2,7,6,5\},\{3,4\}\}$ with hubs 1,2, and 3 are crossbred using Double-Cluster Exchange Crossover (DCEC). The detailed steps are given below.

1. Two randomly selected clusters, $C_2P_1 = \{\mathbf{4},5,8,9\}$ from P_1 and $C_2P2 = \{\mathbf{2},7,6,5\}$ from P_2 respectively, are swapped between P_1 and P_2 producing offspring Ch_1 and Ch_2.

2. If an incoming cluster has the same hub as a home cluster, both are merged. This step doesnt apply in the present case.

3. Nodes (4,8,9) of the incoming cluster C_2P_1 to Ch_2 are also contained in other clusters of Ch_2. They are deleted from the rest of the clusters but retained in C_2P_1. If a node in an incoming cluster is a hub in another cluster, it is deleted from the incoming cluster but retained in the other cluster. The same process is repeated for Ch_1.

4. The missing nodes (2,6,7) are re-inserted in Ch_2 based on distance. A node is assigned to a hub that is the closer and has sufficient capacity to handle the additional flow from the newly added node. If no such hub can be found, then it is assigned to the nearest hub regardless of the violation of hub capacity. The same step is repeated for Ch_1.

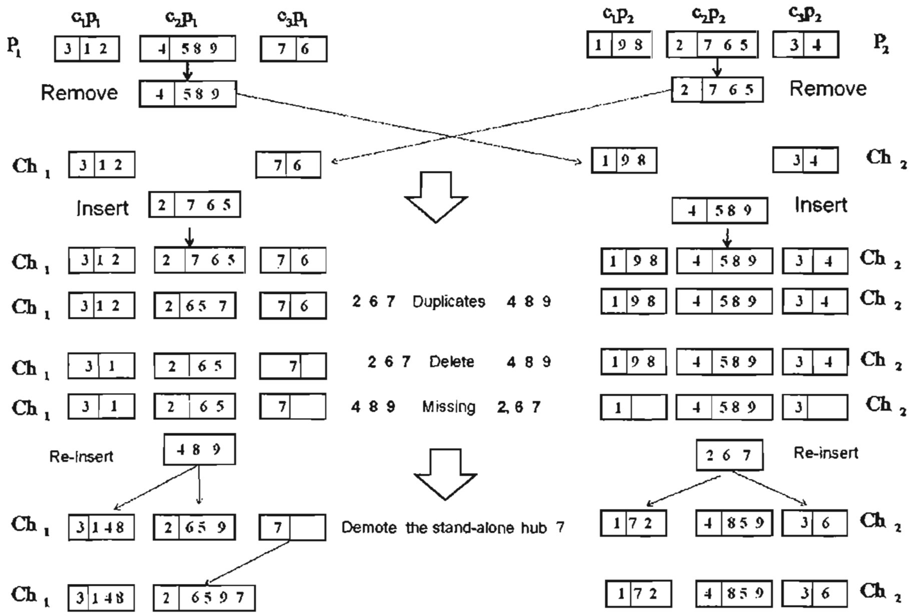


Figure 3.11: Double-Cluster Exchange Crossover

5. The stand-alone hub 7 is removed from Ch_1 and re-assigned to hub 2 of Ch_1 as a spoke using the distance-based assignment rule i.e., a node is assigned to the closest hub.

6. Steps 1 through 4 are repeated a second time to exchange another cluster between the two solutions.

3.4.3 Best Cost Route Crossover(BCRC)

Another crossover used in this work is an adaptation of the Best Cost Route Crossover (BCRC). This crossover was introduced by Ombuki *et al.* [23] for vehicle routing problems and, subsequently, has been used by many studies on similar problems with good results [23]. The crossover selects a route from one parent and injects elements from it at best possible locations in the opposite parent preserving the feasibility of the parent solution.

The notion of route in a routing problem has close equivalence in a cluster in hub location problems. The BCRC for the SAHLP swaps clusters across the mating parents. This is done by removing a randomly selected cluster from one parent and shifting its contents to the other parent. Same action is performed on the second parent. The hubs replace each other in the destination parents and the nodes are inserted (in the destination parents) based on distance.

The BCRC crossover for the single allocation hub location problem is illustrated in Figure 3.12. In the figure, two parent solutions P_1 and P_2, based on a problem instance of size 9 nodes are selected from the population from which two children Ch_1 and Ch_2 are produced using BCRC. Parent P_1 represents a hub-spoke network with clusters $C_1P_1 = \{3, 1, 7\}$, $C_2P_1 = \{5, 6, 9\}$, and $C_3P_1 = \{4, 2, 8\}$ and P_2 , $C_1P_2 = \{1, 4, 5\}$, $C_2P_2 = \{2, 9, 6, 8\}$, and $C_3P_2 = \{7, 3\}$. The recombination operation works as follows.

1. Two random clusters e.g. C_2P_1 and C_2P_2 are selected from P_1 and P_2 respectively. The nodes in C_2P_1 are deleted from P_2 and, conversely, nodes in C_2P_2 are deleted from P_1.

2. The hub i.e. 5 in cluster $C_2P_1 = \{5, 6, 9\}$ is inserted in parent solution P_2 as the new hub. The same operation is repeated for $C_2P_2 = \{2, 9, 6, 8\}$ and parent P_1.

3. Next, nodes from C_2P_1 i.e. 6 and 9 are inserted in P_2 in appropriate clusters using distance assignment rule and based on the non-violation of hub capacity. If no cluster can be found for a node without violating the hub

capacity, then it is assigned to the closest hub. Same operation is repeated for C_2P_2 and P_1.

4. Stand-alone hubs in the children solution are demoted and assigned to other hubs as spokes.

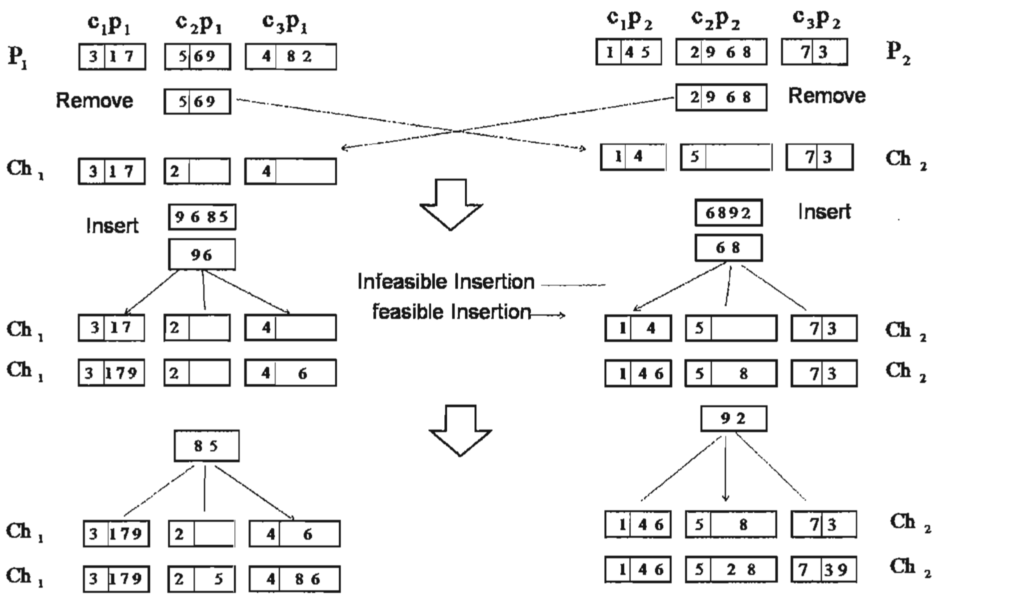


Figure 3.12: Best-Cost Routing Crossover

3.5 Mutation

The mutation operations employed in the proposed GA are *Shift Node*[3][4], *Swap Nodes*[3][4], and *Replace-Hub* mutations. The shift and swap mutation operations can be performed only for solutions with multiple clusters. In the *Shift Node* mutation, a node is detached from one cluster of a solution and inserted into another cluster of the solution. The node to be shifted is selected randomly. Likewise, the source and destination clusters for the node are choosen randomly. The shift mutation operation can be performed only for clusters with more than one node.In *Swap Node* mutation, two clusters are randomly selected from the given solution and one random node from the first cluster is shifted to the second cluster. Likewise, from the second cluster, a randomly selected node is shifted to the first cluster.In the *Replace-Hub* mutation, the hub from a randomly selected cluster of a solution is demoted as spoke whereas a spoke from the same cluster is promoted as hub. The operations are illustrated in the Figure 3. 13.

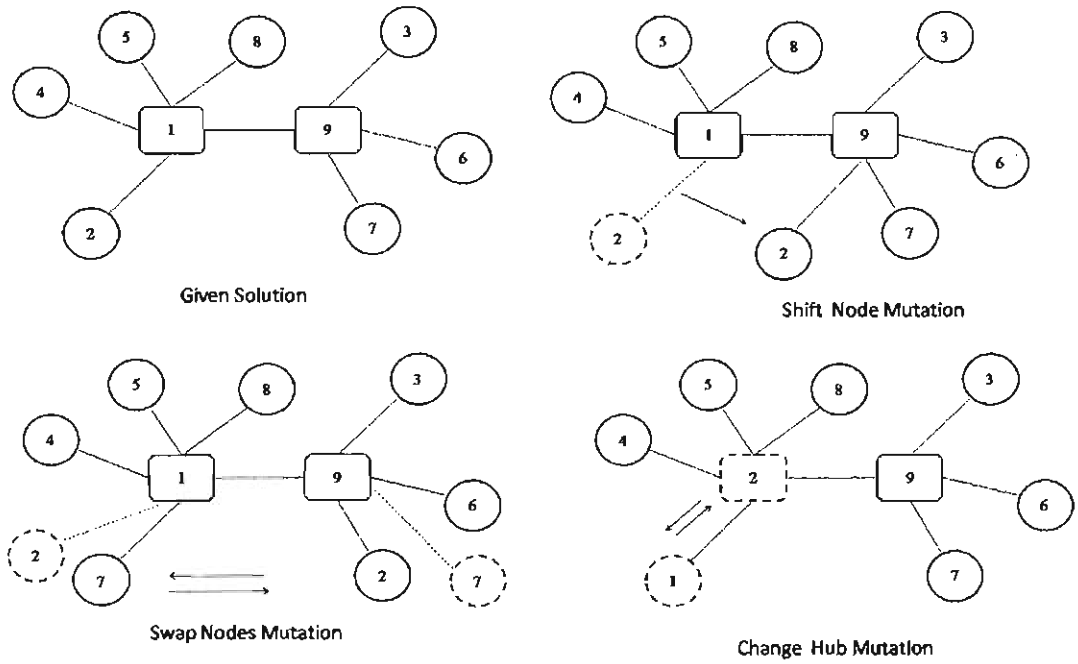


Figure 3.13: Mutation Operations

3.6 Constraint Handling in GA-SAHLP

There are six constraints that apply to the Single Allocation Hub Location Problem as given in the CSAHLP-H formulation in section 2.1. Constraints (1) and (2) enforce the single-assignment restriction. Constraints (3) and (4) i.e., capacity constraints, put an upper bound on the incoming flow to a hub in the capacitated SAHLP. Constraint (5) enforces integrity and constraint (6) ensures that flow from a spoke is routed only through a hub.

In the proposed GA, constraints, (1), (2), (5) and (6) are implicitly handled by different components of the GA. For example, during the initial population generation and crossover operations, a spoke is assigned only to a single hub and duplicate nodes are deleted thus enforcing the single-assignment constraints (1) and (2). These measures also enforce the hub-routing constraint (6) and integrity constraint (5).

Implementation of hub-capacity constraints (3) and (4) is more complex and requires efficient handling. Hub-capacity constraints make sure that the sum of the incoming flow, i.e. the flow from spokes in a cluster to the hub including the flow generated by the hub itself, doesn't exceed the hub capacity. Hub-capacity constraints are violated by the crossover and mutation operations in the GA. In the proposed approach, capacity violation is handled in two ways; through a *pro-active* measure and a *reactive* measure.

In *pro-active* measure, a node is assigned in such a way during the initial population generation or crossover that the capacity of the hub is not violated. This is done by allocating a node to hub that has the capacity. However, feasible insertion of a node sometimes requires reshuffling of clusters, which is computationally expensive. Therefore, a *reactive measure* is adopted by including a special *repair module* in the GA. Repair module adjusts capacity overflow in hubs through careful re-assignment of nodes in a solution.

The repair module is invoked after the creation of a solution during the initial population generation or the crossover operation. Crossover operation first attempts to preserve the feasibility of an offspring by assigning nodes to hubs with capacity. If feasible assignment of a node is not possible, the node is assigned to a hub regardless of the violation of and hub capacity ang a special flag is set. The flag serves to activate the repair module.

An outline of the repair module is given in Algorithm 5. The example in Figure 3.14 gives its illustration. The figure shows node-flows and hub-capacities in an offspring solution. For example, spoke 9 sends flow of 0.5 units via hub 1, which itself generates a flow of 0.3 units. The total incoming flow through hub 1 is the sum of flows from its allocated spokes and the flow it generates itself i.e., the total through hub 1 is 0.5 + 0.4 + 0.2 + 3 = 1.4. As this exceeds the capacity of hub 1 i.e., 1.3, so the cluster containing hub 1 has overflow. The other overflow cluster in the network is the one with hub 5.

To repair the network, nodes 4 and 3 are detached from existing hubs and re-assigned to other hubs in the network. Thus node 3 is re-assigned to hub 11 and node 4 to hub 1 as shown in the figure.

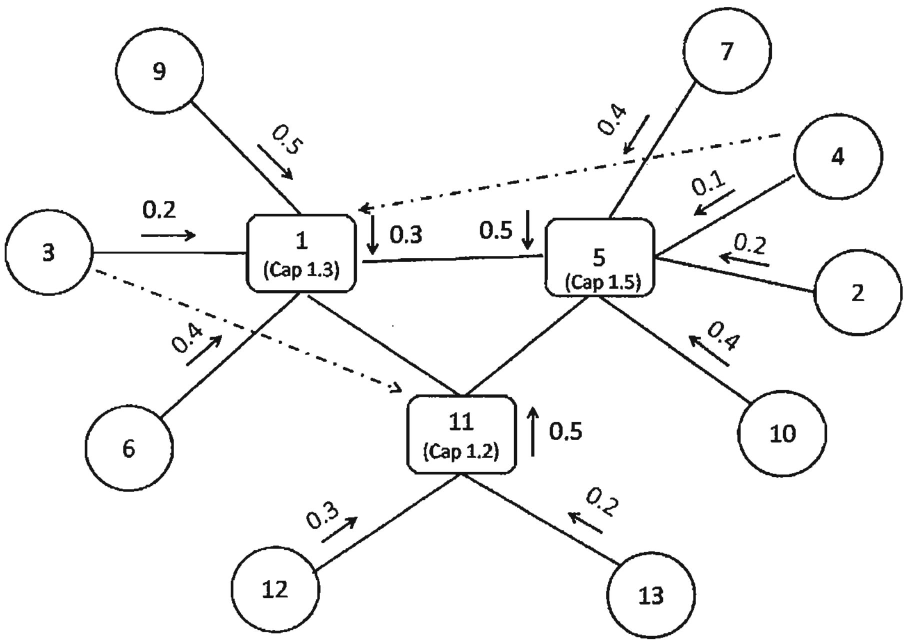


Figure 3.14: An instance of flow through the network

The repair work is carried out in the following steps.

Algorithm 5 Chromosome Repair

---

1. Find an overflow cluster in the given solution.
2. Keep detaching minimum-flow nodes from the cluster until the overall cluster-flow is within the limits of hub-capacity. Save the nodes in an *overflow node list*.
3. Repeat steps 1 and 2 until there is no cluster with overflow.
4. Retrieve the detached nodes from the *overflow node list* one at a time and assign them to the existing hubs. Repeat this step until all the nodes have been assigned or further assignment is not possible without violating the capacities of the hubs.
5. If there are still some un-assigned nodes in the *overflow node list*, create a new hub (from one of the spokes) with sufficient capacity and assign it the remaining nodes.

Detaching nodes with minimum-flow from an overflow hub contributes to the fitness of solutions because value of the objective function in the case of SAHLP depends on flow-volume and distance between nodes. Assigning nodes with larger flows to nearest hubs will minimize the objective function value. Detaching a node with larger flow from a hub during the repair process will, therefore, affect the quality of the solution. Thus, spokes with minimum flows are detached from an overflow hub.

Chapter 4

Computational Results for the Capacitated Single Allocation Hub Location Problem(CSAHLP)

4.1 Experimental Parameters and Setup

The GA proposed in chapter 3 was coded in Java and run on Pentium 4, 1.5 GHz PC in a Windows 2007 environment. To evaluate the computational effectiveness of the proposed GA, an empirical study with three versions of the GA i.e., GA-1, GA-2, and GA-3 was performed. Each version was based on one of the crossovers introduced in section 3.4 using both List-based and Set-based representations as given in section 3.1. The GA versions are given below.

1. GA-1: Double-Cluster Exchange Crossover (DCEC) based GA.
2. GA-2: Multi-Cluster Exchange Crossover (MCEC) based GA.
3. GA-3: Best-Cost Routing Crossover (BCRC) based GA.

Obtaining good parameter settings for a given problem is a crucial factor in the performance of a GA. It is now generally accepted that optimum parameter settings may be problem specific, implying that the GA being designed must first be parameterized in the context of a particular problem [23]. The parameter settings for the GAs i.e., GA-1, GA-2, and GA-3, shown in Table 4.1 were empirically established (Appendix A contains results for crossover rate of 0.80). The experiments were based on 30 runs of each of the above GA versions. The execution of the GA was terminated after 1000 generations or when there was no change in fitness for 150 generations.

Table 4.1: Experimental parameters

Parameter	GA-1	GA-2	GA-3
population size	500	500	500
population	generational	✓	✓
chromosome initialization	random	✓	✓
generational span	1000	✓	✓
probability of crossover	0.55	0.60	0.60
probabliity of mutation	0.2, 0.4	0.4,0.2	0.4, 0.2

Furthermore, after a mutation decision was made according to the mutation rate in the table above, probabilities of 0.2, 0.6, and 0.2 were used to select one of the mutation types i.e., *shift*, *swap*, or *replace-hub* mutation respectively, (Figure 3.14) to apply. A comparative study between the three versions of the GAs is done in terms of solution quality, number of problems solved compared to current best, and computational time. Similarly, a comparative study between the GAs and other published works using GA-based and non-GA based methods is provided.

4.2 Data Sets

Experiments for the CSAHLP were performed using Australian Post (AP) data [1], which is described below.

4.2.1 AP Data

The AP data set was introduced by Ernst *et. al.* [1] and is based on a real application to postal delivery system in Australia. AP data is the only data benchmark data set available for capacitated hub location problems. It has also been used by some studies for uncapacited problems [4][11].

The set contains problems of up to 200 nodes with each node representing a postal district. The problem sizes are 10, 20, 25, 40, 50, 100, and 200 nodes. The internodal flows in AP data set are asymmetric i.e., $W_{ij} \neq W_{ji}$. The data set contains hub costs and hub capacities for capacitated hub location problems. The unit collection and distribution costs i.e., χ and δ, in the data set are 3.0 and 2.0 respectively and the discount factor, α, is 0.75.

There are two types of hub costs and capacities i.e., loose (L) and tight (T) in AP data set. Loose cost/capacity hubs are assumed to have less variation in cost/capacities and tight cost hubs have more variation. By combining these two types of hub costs and capacities, 4 types of problems with different complexity can be derived for each problem size. Tight costs/capacities problems tend to be more difficult than loose cost/capacities problems.

In this work, the type of an AP problem is specified with the notation nFC where n is the number of nodes in the problem, F is the cost type for hub, and C, the capacity type for hub. For example $100LL$ means problem with 100 nodes (i.e., $n = 100$) and loose (L) cost, loose (L) capacity hubs.

4.3 Experimental Results and Discussions for the Capacitated SAHLP

Tables 4.2 and 4.3 together with figures 4.1 and 4.3 present computational results for the GAs i.e., GA-1, GA-2, and GA-3, based on List-based and Set-based representations respectively. Both tables have the same structure. For example, in Table 4.2, the first column gives the name of the problem instance. The column labelled *known-best value* lists the known-best solutions to the problems. For a small-sized problem i.e., $n = 10, 20, 25, 40$, a known-best solution represents the optimal cost as established in [1]. For large-sized problems i.e., problems with $n = 50, 100, 200$, the known-best solution represents best cost obtained by current methods. The next three columns list the best solutions found by GA-1, GA-2, and GA-3 respectively in 30 runs. In the tables and figures, solutions are labelled either *known-best*, *new best*, or *comparable* depending on their quality relative to the current best solutions. The same terms and notations have been used throughout this thesis to designate/denote solutions. These labels and notation have the following meaning.

- A *known-best* solution is the current best solution for a problem as reported in literature on the SAHLP. It is denoted by $\surd$.
- A *new best* solution is a new best solution for a problem. It is represented by a value in bold text.
- A *inferior* solution is a solution that is inferior to the "known-best" solution. It is denoted by a non-bold value.

The figures in this chapter gives a summary of the performance comparison. Detailed comparison can be found in Appendix B. Table D.1, Appendix D give averages of the solution values found in 30 runs of the GA.

4.3.1 Performance Comparison of GA-1, GA-2, and GA-3 with Best Known Solutions using Set-based Representation

Figure 4.1 gives performance comparison between GA-1, GA-2, and GA-3 in terms of known-best or new best solutions found using set representation. As can be seen from the figure, GA-1 obtained 26/28 best solutions, which include two new best solutions. GA-2 and GA-3 obtained 25/28 and 17/28 best solutions respectively including a new best solution for each GA. Thus, GA-1 performed better than GA-2 and GA-3 in terms of the number of known or new best solutions found. Performance of GA-2 was comparable with that of GA-1 whereas GA-3 was the least successful.

Table 4.2 compares solutions found by the GAs with known-best solutions. For purpose of better comparison, the problems have been assigned three categories i.e., the *small-sized* problems with 50 or less nodes ($n \leqq 50$), *medium-sized* problems with number of nodes between 50 and 100 ($n > 50$ and $n \leqq 100$), and *large-sized* problems with more than 100 nodes.

As can be seen from table 4.2, GA-1 and GA-2 were able to solve all *small-sized* problems to optimality as against GA-3, which obtained suboptimal values for the 10LL, 25TL, 40TT and 50TT problems. GA-1 and GA-2 were equally better than GA-3 on small-sized problems.

For the *medium-sized* problems i.e., the 100LL, 100LT, 100TL, and 100TL problems, GA-3 found only comparable values except the 100LT case for which it obtained a new best solution. GA-1 and GA-2 found two known-best and one new best solution each for the medium sized problems. Thus, GA-1 and GA-2 have comparable performance on the medium-sized problems.

As for the *large-sized* 200LL, 200LT, 200TL, and 200TT problems, GA-3 again failed to find any known-best or new best solution. GA-1 obtained new best solution for the 200LT problem and known-best solutions for the 200LL and 200TL problems. Compared to this, GA-2 found known-best solutions for two of the large-sized problems (i.e. the 200LL and 200TL cases) and comparable values for the remaining two problems. GA-1, therefore, performed better than GA-2 on

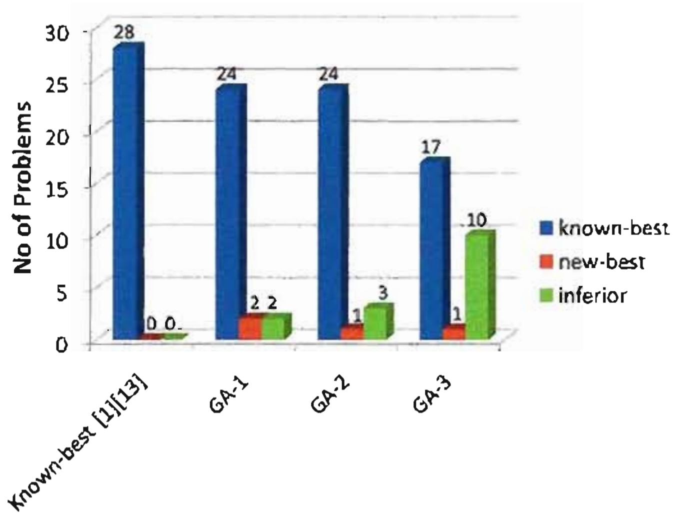


Figure 4.1: CSAHLP: Comparison in terms of the number of known-best or new best solutions using set representation

large-sized problems.

It can thus be concluded for the set representation that GA-1 based on the Double-Cluster Exchange Crossover (DCEC) was the most successful on all types of problems. GA-2 with Multi-Cluster Exchange Crossover (MCEC) performed better on small as well as medium-sized problems but had modest success on large-sized problems. Finally, GA-3, which incorporates the Best-Cost Routing Crossover (BCRC), had some modest success only on small-sized problems. Next the computational times of the GAs are discussed.

Table 4.2: CSAHLP: Computational performance of GA-1, GA-2, and GA-3 using set-representation

Problem	Known-best SA-RDH [1]	GA-1	GA-2	GA-3
10LL	224250.05	√	√	224706.29
10LT	250992.26	√	√	√
10TL	263399.94	√	√	√
10TT	263399.94	√	√	√
20LL	234690.94	√	√	√
20LT	253517.40	√	√	√
20TL	271128.18	√	√	√
20TT	296035.40	√	√	√
25LL	238977.95	√	√	√
25LT	276372.50	√	√	√
25TL	310317.64	√	√	310493.20
25TT	348369.15	√	√	√
40LL	241955.71	√	√	√
40LT	272218.32	√	√	272455.80
40TL	298919.01	√	√	√
40TT	354874.10	√	√	√
50LL	238520.59	√	√	√
50LT	272897.49	√	√	√
50TL	319015.77	√	√	√
50TT	417440.99	√	√	418269.90
100LL	246713.97	√	√	246755.13
100LT	256207.52[13]	256250.32	**256183.42**	**256183.43**
100TL	362950.09	√	√	365247.39
100TT	474670.32	**474660.51**	474667.32	478937.94
200LL	241992.97	√	√	241993.97
200LT	268894.41	**268661.14**	269494.09	272089.92
200TL	273443.81	√	√	273502.88
200TT	291830.66[13]	291969.46	291973.07	292154.47
No of best solutions found		26/28	25/28	17/28

Behaviour of the GAs in terms of computational time versus problem size for set-representation is given by Figure 4.2. As the computational times of the GAs have the most variation for larger problems, therefore, only problems of sizes $n \geqslant 40$ have been considered. The computational time (CPU time) of a GA for a given problem of size "n" (e.g. 200 nodes) has been obtained by adding its average computational times [1] for nLL, nLT, nTL, and nTT (e.g. 200LL, 200LT, 200TL, and 200TT) and dividing the sum by 4.

The figure shows that all the GAs i.e. GA-1, GA-2, and GA-3, have low (below 100 seconds) and almost comparable computational times for small-sized problems. However, for the medium-sized problems, computational times of GA-1 and GA-2 rise sharply although the increase in the computational time of GA-1 is relatively larger than that of GA-2. Computational time of GA-3, on the other hand, remains low for medium-sized problems. However, for large-sized i.e., 200 nodes problems its computational time rises steeply. Against this, computational times of GA-1 and GA-2 remain lower for large-sized problems. On the whole, the behaviour of GA-2 in terms of computational time is better than the other two GAs.

The sudden rise in the computational time of GA-3 for large-sized problems may be due to the reason that the large-sized problems have larger clusters and so GA-3, which is based on the Best-Cost Routing Crossover (BCRC), has to re-assign larger number of nodes when clusters are swapped between the mating parents. Exacerbating this, may be larger instances of hub capacity violation in the case of BCRC due to the re-assignment of large number of nodes, which entails a time-consuming solution-repair work.

As illustrated by the above comparison, GA-1 with Multi-Cluster Exchange Crossover (MCEC) and GA-2 with Double-Cluster Exchange Crossover (DCEC) are better than GA-3 based on Best-Cost Routing Crossover (BCRC) on set representation in terms of the number of problems successfully solved, quality of solutions, and computational time behaviour. Further comparison of the GAs with published works, will, thus, focus on GA-1 and GA-2 only.

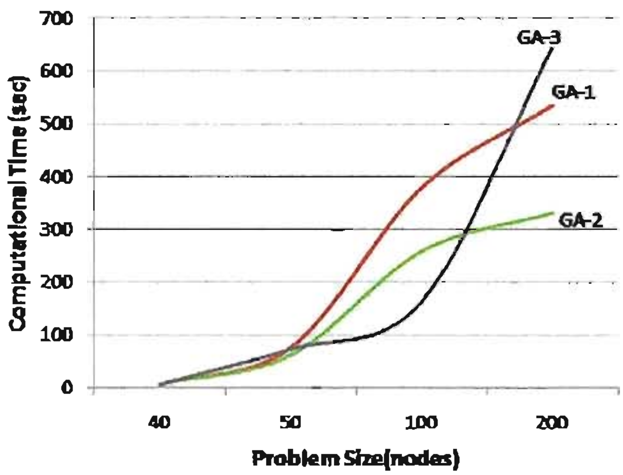


Figure 4.2: CSAHLP: Computational times of GA-1, GA-2, and GA-3 versus problem-size using set representation

4.3.2 Performance Comparison of GA-1, GA-2, and GA-3 with Best Known Solutions using List-based Representation

Figure 4.3 gives performance comparison of the GAs for the list-based representation in terms of best solutions found. As can be seen from the figure, GA-2 found 27/28 best solutions of which 3 were new best solutions. This was followed by GA-1 and GA-2, which obtained best solutions for 23/28 and 17/28 problems respectively. Thus GA-2 was the most successful on list representation in terms of the number of problems solved successfully. GA-3 on the other hand was the least successful.

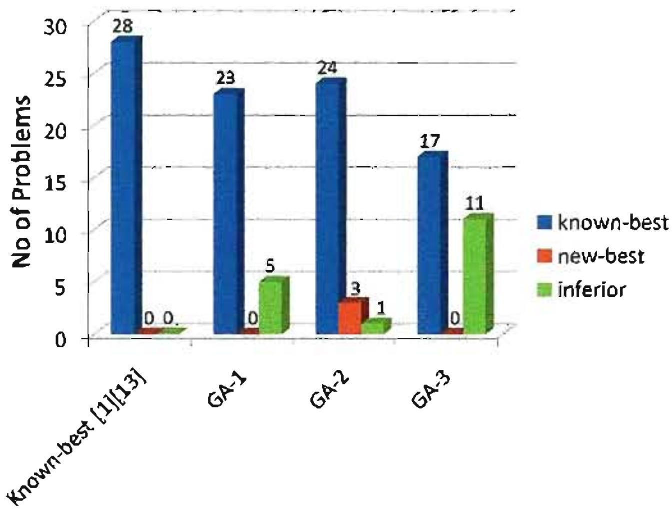


Figure 4.3: CSAHLP: Comparison in terms of the number of known-best or new best solutions found using list representation

Comparison of the solution values obtained by the GAs with the known-best solutions is presented using Table 4.3. The same categorization of the problems into small-sized, medium-sized and large-sized is used to grade the performance of the three GAs on the list representation in terms of solution quality.

As can be seen from the table, GA-1 and GA-2 found optimal or known-best solutions to all the small-sized AP problems. GA-3, on the other hand, failed to obtain optimal solutions for the 10LL, 25TL, 40LT, and 50TT problems. Both GA-1 and GA-2, thus, outperformed GA-3 on the *small-sized* AP problems and the list representation.

On the medium-sized 100LL, 100LT, 100TL, and 100TT problems, GA-2 using list representation performed very well. It found best solutions to all the medium-sized problems including the new best solutions for the 100LT and 100TT instances. GA-1, on the other hand, obtained known-best solutions for two problems and an improved solution relative to the current-best for the 100TT problem. For the 100LT problem, its solution is comparable with the known-best. Solutions of GA-3 for the medium-sized problems are comparable to the known-best. Thus on the *medium-sized* problems, GA-1 had the best performance followed by GA-2.

On the *large-sized* 200LL, 200LT, 200TL, and 200TT problems, performance of GA-2 was again better. It found a new best solution for the 200LT problem and known-best solutions for the 200LL and 200TL problems. For the 200TT problem, its solution value is comparable with the known-best. Against this, GA-1 found known-best solution to one problem i.e. the 200LL case. For the 200LT problem, its solution value is better compared to the known-best but inferior to the solution by GA-2. For the remaining two problems, it obtained comparable values. The solutions obtained by the GA-3 for all large-sized problems are inferior compared to the known-best.

Thus the performance of GA-2 with the Multi-Cluster Exchange Crossover (MCEC) performed better for all types of problems while GA-1 based on the Double-Cluster Exchange Crossover (DCEC) gave comparable performance with GA-2 on small and medium-sized problems. Last, GA-3, which incorporates the the Best-Cost Routing Crossover (BCRC), showed some modest performance on small-sized problems only. GA-2 with MCEC thus had the best performance for AP problems on the List representation. Next the behaviour of the GAs in terms of computational time is discussed.

A comparison of the average computational times of the GAs for the list representation is given in Figure 4.4. As the figure shows, computational time of GA-3

Table 4.3: CSAHLP: Computational performance of GA-1, GA-2, and GA-3 using list-representation

Problem	Known-best SA-RDH [1]	GA-1	GA-2	GA-3
10LL	224250.05	√	√	224706.29
10LT	250992.26	√	√	√
10TL	263399.94	√	√	√
10TT	263399.94	√	√	√
20LL	234690.94	√	√	√
20LT	253517.40	√	√	√
20TL	271128.18	√	√	√
20TT	296035.40	√	√	√
25LL	238977.95	√	√	√
25LT	276372.50	√	√	√
25TL	310317.64	√	√	310493.20
25TT	348369.15	√	√	√
40LL	241955.71	√	√	√
40LT	272218.32	√	√	272455.80
40TL	298919.01	√	√	√
40TT	354874.10	√	√	√
50LL	238520.59	√	√	√
50LT	272897.49	√	√	√
50TL	319015.77	√	√	√
50TT	417440.99	√	√	418269.90
100LL	246713.97	√	√	246755.13
100LT	256207.52[13]	256250.32	**256155.33**	256455.24
100TL	362950.09	√	√	364515.45
100TT	474670.32	474287.49	**474184.94**	476568.42
200LL	241992.97	√	√	242776.31
200LT	268894.41	268487.45	**267827.979**	269276.97
200TL	273443.81	273541.82	√	273502.88
200TT	291830.66[13]	292237.69	291891.69.07	292528.72
No of best solutions found		23/28	27/28	16/28

remains low for small and medium-sized problems but rises steeply for large-sized problems (i.e. problems with more than 100 nodes). There is also a sharp rise in the computational time of GA-1 for medium-sized problem until the 200 nodes problem when its computational time drops slightly. The computational time for GA-2 rises comparatively slowly with increase in the problem size. GA-2, thus, has better behaviour than either GA-1 or GA-3 in terms of computational time. GA-3 is better than GA-1 and GA-2 for small and medium-sized problems but is more expensive for large-sized problems.

The comparison of GA-1, GA-2, and GA-3 indicates that GA-2 based on the Multi-Cluster Exchange Crossover (MCEC) has the best performance on list representation. GA-1 incorporating the Double-Cluster Exchange Crossover (DCEC) has matching performance with GA-2 in terms of solution quality although it finds fewer solutions that are the known or new best. Further comparisons of the GAs with published work on list representation, therefore, will focus only on GA-1 and GA-2.

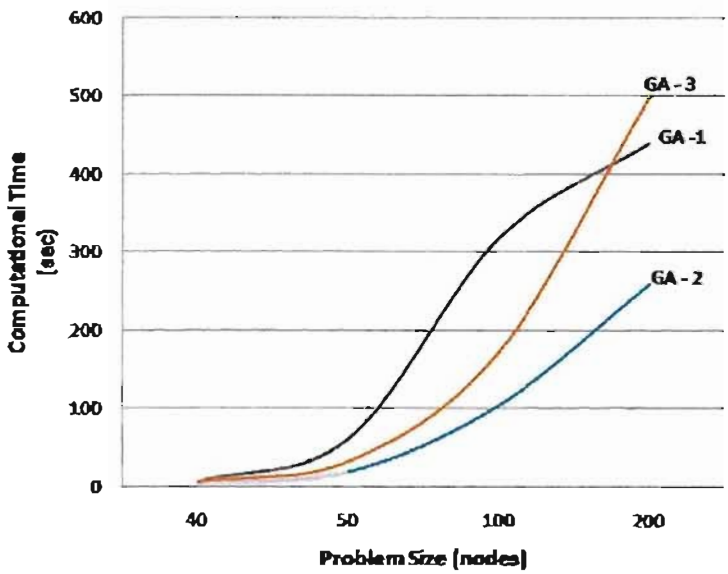


Figure 4.4: CSAHLP: Computational times of GA-1, GA-2, and GA-3 versus problem size using list representation

4.3.3 Performance Comparison of GA-1 and GA-2 with Published GA Approaches

Performance comparison of GA-1 and GA-2 for the two representations i.e., the set and the list representations, with published GA approaches in terms of the number of known-best or new best solutions found is given in the following sections. Comparison in terms of individual solution values can be found in Tables B.1 and B.2, Appendix B.

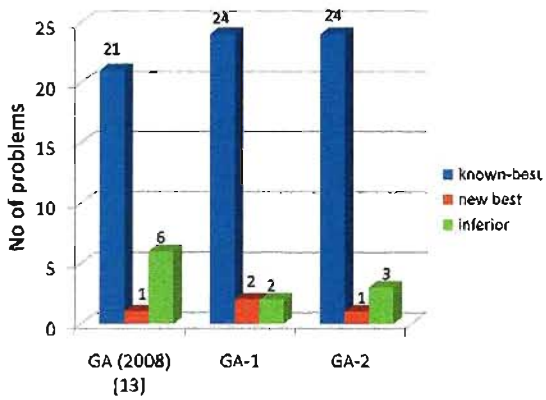


Figure 4.5: CSAHLP: Comparison with published GA approaches using set-representation

4.3.3.1 Set-based Representation

As can be seen from Figure 4.5, both GA-1 and GA-2 performed better than GA (2008) [13] on the set representation. GA-1 obtained 26/28 best solutions including two new best solutions compared to the 21/28 best solutions of GA (2008) [13]. Likewise, GA-2 with 25/28 known and new best solutions also outperformed GA (2008) [13]. Moreover, GA-1 and GA-2 found relatively superior solutions for many problems as can be known from Table B.1 of Appendix B. This includes some small AP problems for which GA (2008) [13] has suboptimal solutions.

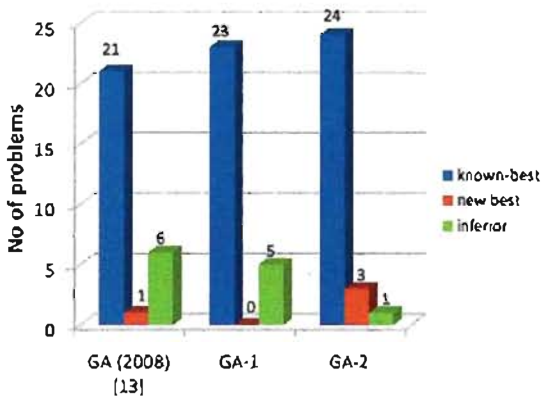


Figure 4.6: CSAHLP: Comparison with published GA approaches using list-representation

4.3.3.2 List-based Representation

The performance comparison of GA-1, GA-2, and GA (2008) [13] for the list representation is given in Figure 4.6. As depicted by the figure, GA-1 and GA-2 with 27/28, including 3 new best solutions, and 23/28 best solutions respectively outperformed GA (2008) [13] with 21/28 best solutions. Moreover, both GA-1 and GA-2 solved some small-sized AP problems to optimality that GA (2008) [13] couldn't solve as can be found from Table B.2, Appendix B. Likewise, for most of the medium and large-sized AP problem, performance of GA-1 and GA-2 using the list representation was significantly better (Table B.2, Appendix B).

4.3.3.3 Conclusion

The comparison of GA-1 based on the Multi-Cluster Crossover (MCEC) and GA-2 based on the Double-Cluster Exchange Crossover (DCEC) with GA (2008) [13] for the capacitated SAHLP presented in this section show the effectiveness of GA-1 and GA-2 relative to GA (2008) [13]. The GAs outperformed GA (2008) [13] on both representations in terms of the number of problems successfully solved and quality of solutions. Next the performance of the GAs is compared with published non-GA work.

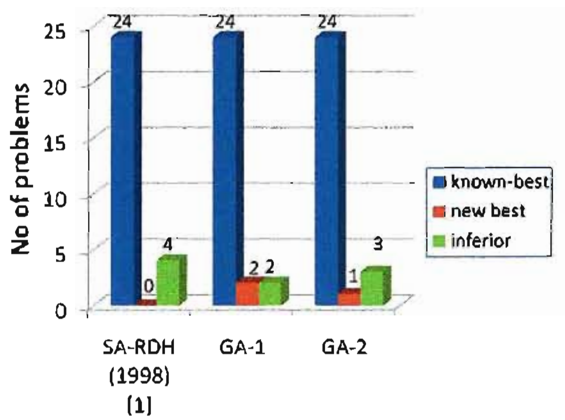


Figure 4.7: CSAHLP: Comparison with non-GA approaches using set-representation

4.3.4 Comparison of GA-1 and GA-2 with Published non-GA Approaches

The next two sections summarise performance comparison of GA-1 and GA-2 in terms of the number of best solutions found with published non-GA approaches to the capacitated SAHLP for both representations. Detailed comparison in terms of individual solution values can be found in tables B.3 and B.4 of Appendix B.

4.3.4.1 Set-based Representation

Figure 4.7 gives performance comparison of GA-1 and GA-2 with the SA-RDH (1998) [1] for set representation. As can be seen from the figure, GA-1 with

24/28 known-best solutions and 2/28 new best compared to the 24/28 solutions of SA-RDH[1] has slightly better performance. Likewise, GA-2, with 25/28 best solutions, including one new best, also outperforms the SA-RDH (1998) [1]. Moreover, whereas performance of GA-1, GA-2, and SA-RDH (1998) [1] is similar on the small-sized AP problems, GA-1 and GA-2 prove to be better than SA-RDH (1998) [1] on some of the medium and large-sized problems, as can be found in Table B.3 of Appendix B.

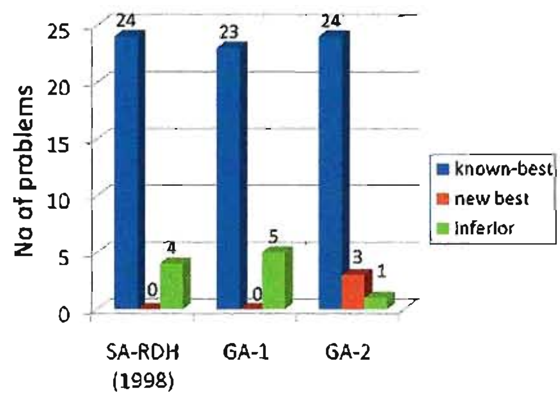


Figure 4.8: CSAHLP: Comparison with non-GA approaches using list representation

4.3.4.2 List-based Representation

The performance summary of GA-1, GA-2, and SA-RDH (1998) [1] is given in Figure 4.8 for list representation. Again, GA-2 with 27/28 best solutions including three new best solutions, as indicated in the figure, outperforms SA-RDH [1998][1], which has 24/28 best solutions. On the other hand performance of GA-1 and SA-RDH (1998)[1] with 23/28 and 24/28 best solutions respectively is similar in terms of the number of problems successfully solved. Furthermore, whereas

GA-1, GA-2, and SA-RDH (1998) [1] have similar performance on small-sized AP problems, on medium and large-sized problems, GA-1 and GA-2 prove to be more effective. Comparison of individual solution values for GA-1, GA-2, and SA-RDH(1998)[1] is given in Table B.4, Appendix B.

4.3.4.3 Conclusion

The comparison of GA-1 and GA-2 with non-GA approaches to the capacitated SAHLP as given above shows the effectiveness of GA-2 based on Multi-Cluster Exchange Crossover (MCEC) and GA-1 based on the Double-Cluster Exchange Crossover (DCEC) relative to the SA-RDH (1998) [1]. Performance of GA-2 (MCEC) was better than the SA-RDH (1998)[1] on both representations whereas the performance of GA-1(DCEC) was better than SA-RDH (1998) [1] on the set representation only.

4.3.5 Comparing Best GAs on the Set and List-based Representations with the Existing GAs

Comparisons in the preceding sections have established GA-1 based on the Multi-Cluster Exchange Crossover (MCEC) and GA-2 based on the Double-Cluster Exchange Crossover (DCEC) to be the best GAs on the set representation and the list representations respectively. Table 4.4, together with Figure 4.9, compares performance of GA-1 using set representation, GA-2 using list representation, and the GA (2008) [13]. As can be seen from the table, GA-2 (list representation) has the best performance of all the three GAs. It finds *known-best* solutions to 24 problems and *new best* solutions to 3 of the 28 CSAHLP problems compared to 24/28 and 21/28 known-best solutions by GA-1 (set representation) and GA (2008) [13] respectively. Its solution value for the 200TT problem is comparable with that of the GA (2008) [13]. The table shows that GA-1 (set representation) also has better performance than GA (2008) [13]. Its solution values for the 100TT and 200LT cases are better than those of GA (2008) [13] and comparable for 100LT and 200TT cases with those of GA (2008) [13].

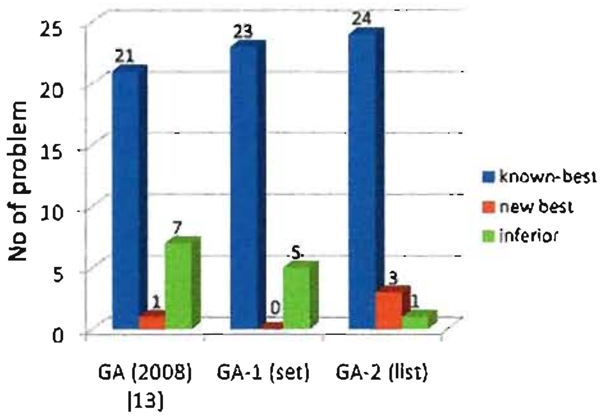


Figure 4.9: CSAHLP: Performance comparison of GA (2008) [13], GA-1 (Set), and GA-2 (List) in terms of number of known or new best solutions

Computational time for GA-1 (set) and GA-2 (list) as a function of problem size has been shown in the Figure 4.10. As the graph depicts, the behaviour of GA-2 (list representation) in terms of computational time is better than that of GA-1 (set-based representation) showing that there is lower increase in the computational time of GA-2 relative to GA-1 with increase in the problem size.

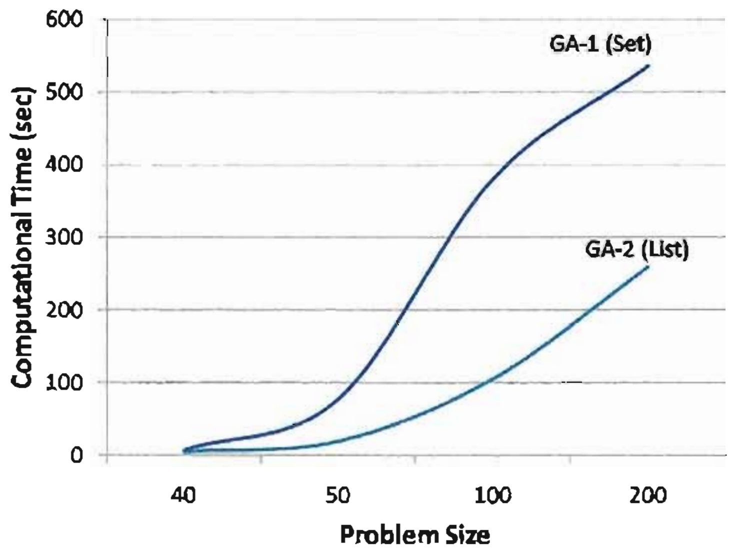


Figure 4.10: CSAHLP: Comparison of computational times for GA-1 (Set) and GA-2 (List)

Table 4.4: CSAHLP: Comparing set and list-based representations, AP data set

Problem	GA(2008) Stanimirovic[13]	GA-1 Set	GA-2 List
10LL	224250.05	√	√
10LT	250992.26	√	√
10TL	263399.94	√	√
10TT	263399.94	√	√
20LL	234690.94	√	√
20LT	253517.40	√	√
20TL	271128.18	√	√
20TT	296035.40	√	√
25LL	238977.95	√	√
25LT	276372.50	√	√
25TL	310317.64	√	√
25TT	348369.15	√	√
40LL	241955.71	√	√
40LT	272218.32	√	√
40TL	298919.01	√	√
40TT	356507.86	**354874.10**	**354874.10**
50LL	238520.59	√	√
50LT	272897.49	√	√
50TL	319015.77	√	√
50TT	422794.56	**417440.99**	**417440.99**
100LL	√	√	√
100LT	256207.52	256250.41	**256155.33**
100TL	364515.24	**362950.09**	**362950.09**
100TT	475156.75	474660.51	**474184.94**
200LL	√	√	√
200LT	270202.25	268661.14	**267827.97**
200TL	273443.81	√	√
200TT	**291830.66**	291969.46	291891.69

4.3.6 Conclusions

Computational results for the proposed GAs i.e., GA-1, GA-2, and GA-3, based on the Double-Cluster Exchange Crossover (DCEC), Multi-Cluster Exchange Crossover (MCEC), and Best-Cost Routing Crossover (BCRC) respectively were presented in this chapter. The results showed that the Multi-Cluster Exchange Crossover (MCEC) employed in GA-2 was effective on the List representation whereas the Double-Cluster Exchange Crossover (DCEC) used in GA-1 was efficient on the Set representation. Furthermore, *Multi-Cluster Exchange Crossover (MCEC) with the List representation* was found to be the most effective of the crossover-representation combinations investigated in this work for the capacitated SAHLP. Comparison with other GA and non-GA methods for the capacitated SAHLP affirmed the efficacy of the crossovers. Overall, the proposed approach was efficient on the given data and consistently produced good-quality results as can be seen from tables 4.5 and 4.6. Investigation with more data may give further insight into the capabilities of these crossovers and solution representations.

Table 4.5: Set representation

	GA-1 (DCEC)	GA-2 (MCEC)	GA-3 (BCRC)
New best	2 / 28	1 / 28	1 / 28
Known-best	24 / 28	24 / 28	17 / 28
Inferior	2 / 28	3 / 28	10 / 28
Overall	26 / 28	25 / 28	18 / 28

Table 4.6: List representation

	GA-1 (DCEC)	GA-2 (MCEC)	GA-3 (BCRC)
New best	0 / 28	3 / 28	0 / 28
Known-best	23 / 28	24 / 28	18 / 28
Inferior	5 / 28	1 / 28	10 / 28
Overall	23 / 28	27 / 28	18 / 28

Chapter 5

Computational Results for the Uncapacitated Single Allocation Hub Location Problem (USAHLP)

5.1 Experimental Parameters and Setup

To evaluate the computational effectiveness of the proposed GA approach for the uncapacitated SAHLP, experiments involving 30 runs of GA-1, GA-2, and GA-3 respectively were performed in the same environment as given in 4.1. The GAs were run on both set and list representations. The same GA parameters as in Table 4.1 were employed. Section 5.2 describes the data sets used in the experiment whereas section 5.3 presents the results.

5.2 Data Sets

In the experiments for the uncapacitated SAHLP, the two standard data sets i.e., Civil Aviation Board (CAB) data set [24] and Australian Post (AP) data set [1] for hub location problems were used. The AP data set [1]has been described in 4.2.1. Section 5.2.1 gives its description for the uncapacitated SAHLP. Section 5.2.2 describes the CAB [24] data set.

5.2.1 AP Data Set

As the hub capacity constraint doesn't hold for the uncapacitated SAHLP, the AP data set [1] for the SAHLP involves two types of problems. The loose-cost (L) problems have less variation in the hub-cost whereas the tight-cost (T) problems have more variation in the hub-cost. Tight-cost (T) problems tend to be more difficult than the loose-cost (L) problems. These two types of problems have been denoted here by the notation nF where n stands for the number of nodes in the problem and F denotes the cost-type i.e., loose, "L" or tight, "T". For example, notation "10L" denotes the loose-cost (L) problem with 10 nodes.

5.2.2 CAB Data Set

The CAB benchmark data set by O'Kelley [24] is based on air traffic between 25 cities in USA. The data set contains test problem instances of 10, 15, 20, and 25 nodes for uncapacitated hub location problems. Unlike the AP data with asymmetric flows between nodes, the internodal flow (W_{ij}) in CAB data set is symmetric i.e., $W_{ij} = W_{ji}$ and is scaled by division with the total network flow i.e.,

$$\sum_{i=1,j=1}^{i=n,j=n} W_{ij}$$

The unit collection cost χ and unit distribution cost δ in the data set are both fixed at 1.0 [3][4][11]. The transfer cost α i.e., the cost for hub-to-hub flow, is varied between 0.2 and 1.0 to provide discount factors for bulk transportation between hubs [3][4][11].

5.3 Experimental Results and Discussions for the Uncapacitated SAHLP Using AP Data

The detailed computational results of the proposed GAs for the Uncapacitated Single Allocation Hub Location Problem (USAHLP) using AP [1] data set are given in tables C.5 and C.6, Appendix C. Means of the best values can be found in Table D.2, Appendix D. Here only summaries of the performance comparison of the GAs i.e., GA-1, GA-2, and GA-3, for the set and list representation are given. Sections 5.3.1 and 5.3.2 give performance comparison between GA-1, GA-2, and GA-3. Performance comparison with published GA and non-GA works is given in sections 5.3.3 and 5.3.4 respectively.

5.3.1 Performance Results for GA-1, GA-2, and GA-3 using Set-based Representation

Figure 5.1 gives the performance comparison of the GAs i.e., GA-1, GA-2, and GA-3, using set representation in terms of the number of known-best or new best solutions. As can be seen from the figure, GA-1 and GA-2 performed equally better by finding 12 known-best solutions each as against 7 known-best solutions of GA-3. Comparison in terms of individual solution values can be found in Table C.5, Appendix C.

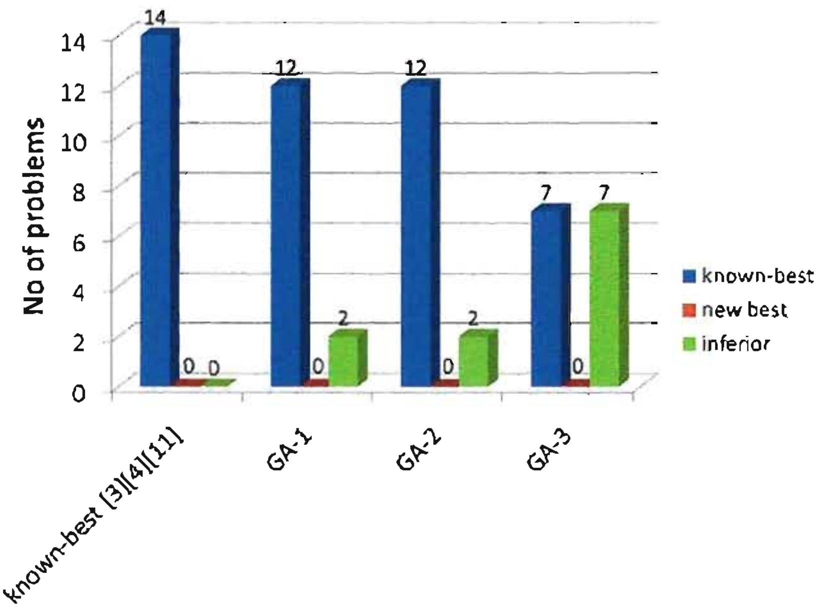


Figure 5.1: USAHLP: Performance of GA-1, GA-2, and GA-3 using set representation and AP data

Figure 5.2 presents average computational time (measured as CPU time) of the GAs as a function of the problems size (i.e. number of nodes, n, in the problem). The computational times have been obtained in the same way as for the capacitated problems. As variation in computational time with the problem size is more apparent in the case of problems with $n \geqslant 40$, so only problems with 40 or more nodes have been considered. Computational times for all the GAs remain low for problems with 50 or less nodes. However, there is relatively higher rise in the computational time of GA-3 when number of nodes exceeds 50. Computational times of all the GAs rise sharply for the 200-node problem. However, the increase in computational times of GA-2 and GA-3 is relatively low compared to the increase in the computational time for GA-1.

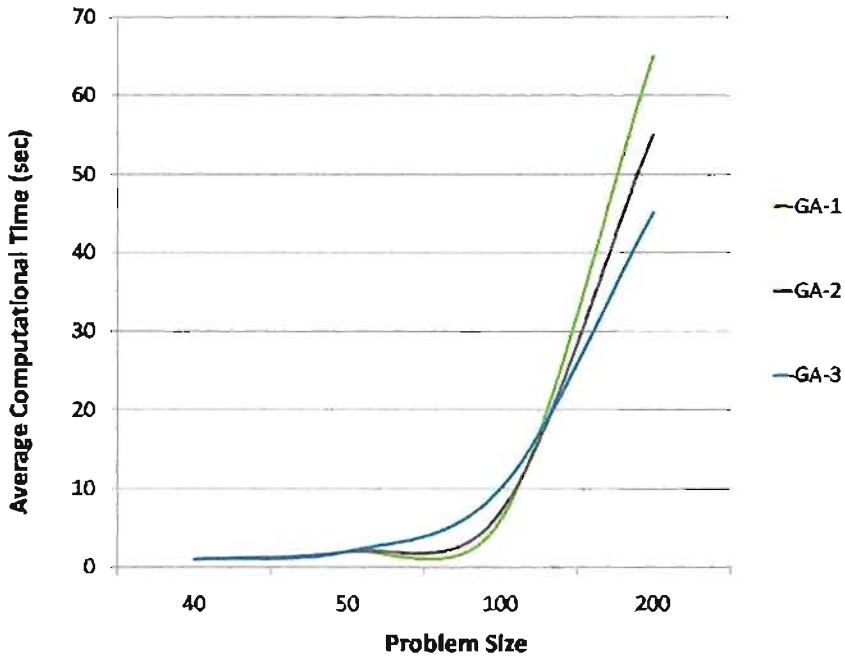


Figure 5.2: USAHLP: Computational time versus problem size, set representation, AP problems

5.3.2 Performance Results for GA-1, GA-2, and GA-3 using List-based Representation

Figure 5.3 gives the performance comparison of the GAs i.e., GA-1, GA-2, and GA-3, using list representation. As in the case of set-representation, GA-1 and GA-2 with 12 /14 known-best solutions gave comparatively better performance than GA-3, which found only 8/12 known-best solutions. The behaviour of GA-1 and GA-2 relative to GA-3 in terms of computational time was also satisfactory as shown by Figure 5.2. GA-3, therefore, will be dropped from further consideration in comparisons with other works, henceforth, and only GA-1 and GA-2 will be focussed upon. Comparison in terms of individual solution values can be found in Table C.6, Appendix C.

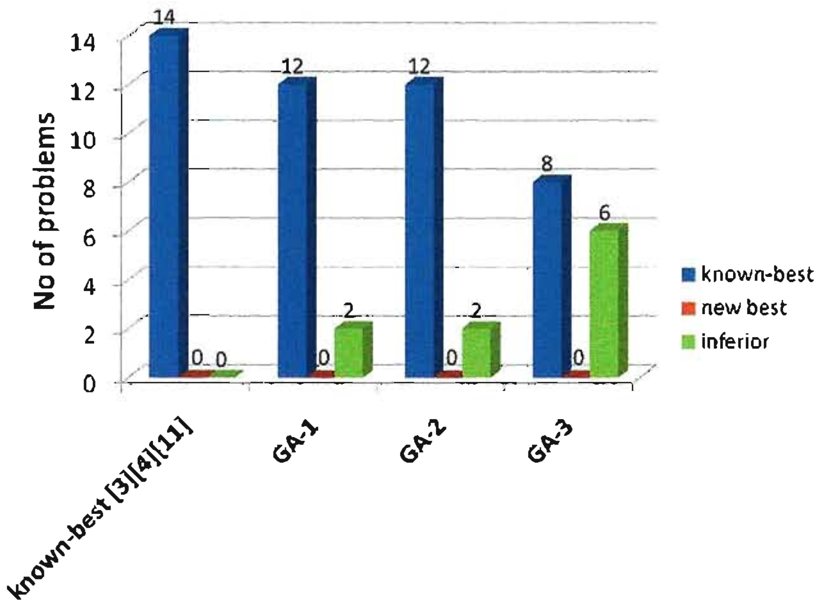


Figure 5.3: USAHLP: Performance of GA-1, GA-2, and GA-3 using list representation and AP data

Computational time of the GAs on list representation as a function of problem size is shown in Figure 5.4. All the three algorithms showed similar behaviour in terms of computational time for both representations i.e., their computational times remain low when the problem has 100 or less nodes but rise sharply for the 200-node problem. Rise in the computational time of GA-3, however, is relatively lower compared to the rise in the computational time of GA-1 and GA-2 for the 200-node problem.

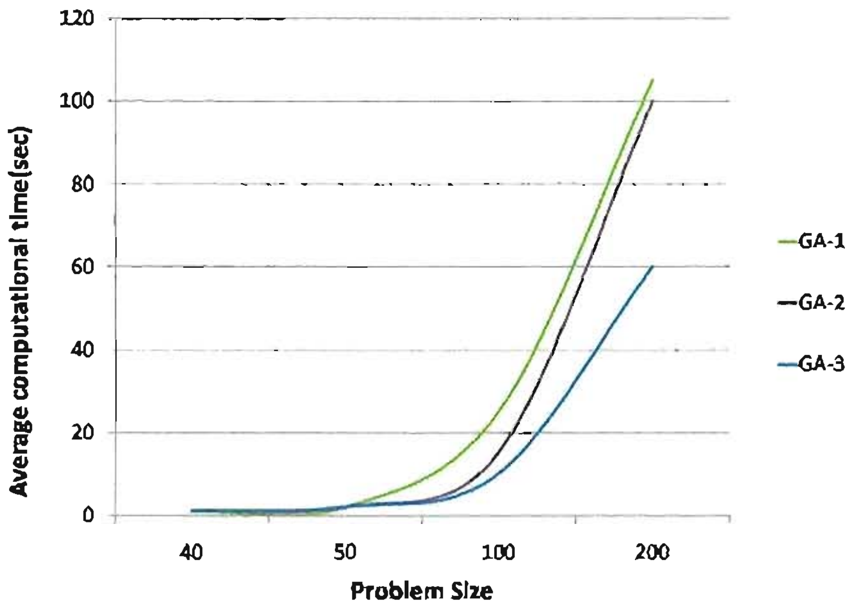


Figure 5.4: USAHLP: Computational time versus problem size using list representation, AP data

5.3.3 Performance Comparison with Published GA Works

As GA-1 and GA-2 have the same performance in terms of the number of known-best or new best solutions found for both the representations, their performance comparison with published GA works is given by the same figure i.e., Figure 5.5. As the figure shows, GA-1 and GA-2 found 12/14 known-best solutions each for both set and list representations as against 7/14 known-best solutions of GATS (1998) [4] and 13/14 known-best solutions of GA (2005) [3]. GA-1 and GA-2 thus performed better than GATS (1998) [4] and almost equally good as GA (2005) [3]. Comparison with published GA works in terms of individual solutions can be found in tables C.1 and C.2, Appendix C.

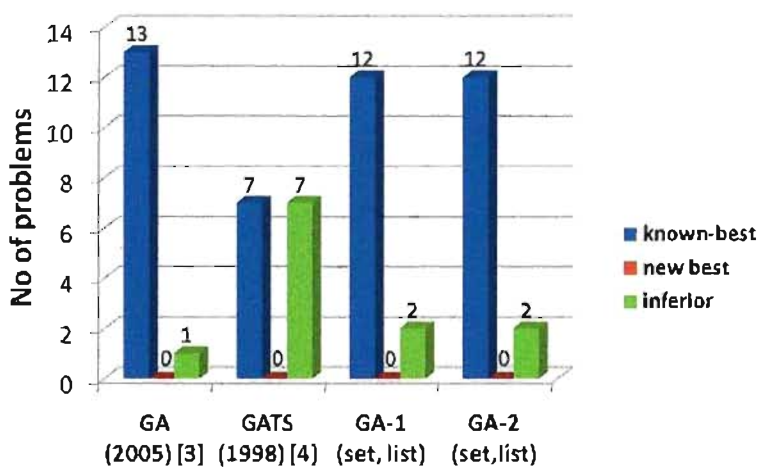


Figure 5.5: USAHLP: Performance comparison of GA-1 and GA-2 using set and list representations with published GA works, AP data

5.3.4 Performance Comparison with Published non-GA Works

Performance comparison of GA-1 and GA-2 in terms of known-best or new-best solutions using both set-based and list-based representations with published non-GA works is given in Figure 5.6. As can be seen from the figure, GA-1 (set and list representations) and GA-2 (set and list representations) with 12/14 known best solutions each have comparable performance with SATLUHLP (2007) [11] with 14/14 known-best solutions and SA (2005) [3] with 13/14 known-best solutions respectively. Comparison with published non-GA works in terms of individual solutions can be found in tables C.3 and C.4, Appendix C.

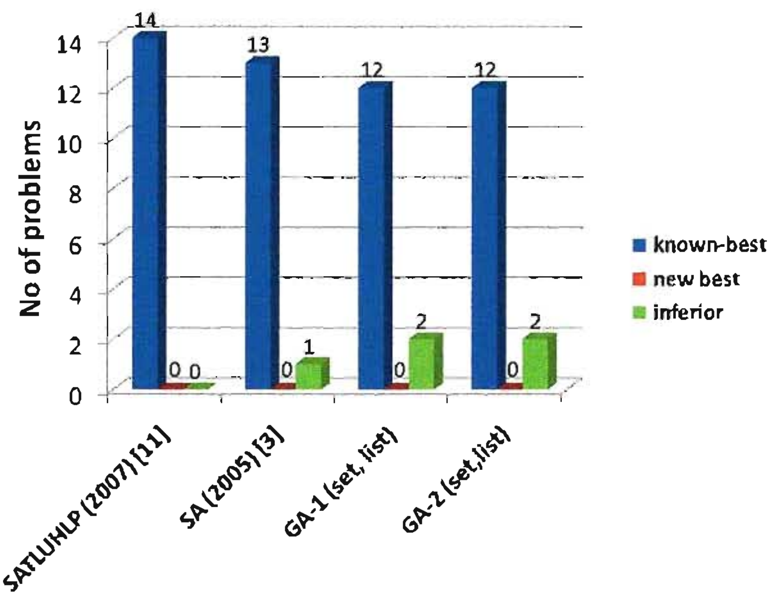


Figure 5.6: USAHLP: Performance comparison of GA-1 and GA-2 using set and list representations with published non-GA works, AP data

Results and comparisons in the preceding sections show that while GA-3 based on Best Cost Routing Crossover (BCRC) was modestly successful on the AP data. GA-1 based on Double-Cluster Exchange Crossover (DCEC) and GA-2 based on Multi-Cluster Exchange Crossover (MCEC) gave same performance on both the representations i.e., Set-based representation and List-based representation. This indicates the effectiveness of both the crossovers i.e., MCEC and DCEC on either representation for the uncapacitated SAHLP. Furthermore, the comparison indicate that, although, performance of both GA-1 and GA-2 is comparable with other methods, there is room for further development to better compete with non-GA methods.

5.4 Discussion and Experimental Results for the Uncapacitated SAHLP Using CAB Data

In the experiments with the CAB data, values of the discount factor (i.e. the *transfer cost*, α) are set to 0.2, 0.4, 0.6, 0.8, and 1.0. *Collection cost* (χ) and *distribution cost* (δ) are maintained at 1.0. The fixed costs for establishing hubs are 100, 150, 200, and 250. In this way, 20 different cases of each problem instance are solved by the algorithm. Total cases solved for all problem instances are thus 80. The optimal solution value is computed as below.

$$\text{Optimal solution value} = min_p \left\{ Opt_p + F \times p \right\}$$

where Opt_p is the optimal solution value for SApHMP given that the number of hubs is p and the fixed cost for establishing a hub F. As the expression indicates, the target solution is obtained by first obtaining solution values of a problem for different number of hubs (p) and then choosing the best value. For example, for problem of size 10 nodes, solution values are obtained for 1, 2, and 3 hubs and the best value is selected as the final solution.

The detailed results of the GAs for the CAB data are presented in tables C.7 through C.14, Appendix C. The first column in each of these tables (ref. Table C.7) specifies the discount α (i.e. the transfer cost), the second column gives the hub cost, the third column lists the optimal value, and the next column gives the optimal hub combination for a given problem. The remaining columns list the solution values for the existing solution approaches and the proposed GAs. Means of the best values of 30 runs can be found in tables D.3 through D.6, Appendix D.

5.4.1 Computational Results Using Set-based Representation

Figure 5.7 summarise performance of the GAs for set representation. A total of 80 CAB problems were studied. GA-2 solved all the 80 CAB problems to optimality. Thus, its performance is matching that of SATLUHLP (2007) [11] and better than that of the GA (2005) [3] and GATS (1998) [4] as can be seen from the figure. GA-1 solved 79/80 problems. For the remaining one problem, its solution was comparable with the optimal value. GA-3 obtained optimal solutions to 72/80 problems and comparable solutions to the remaining problems. Detailed results are given in tables C.7 through C.10, Appendix C. Mean of the best values can be found in Table E.3 through E.6, Appendix E

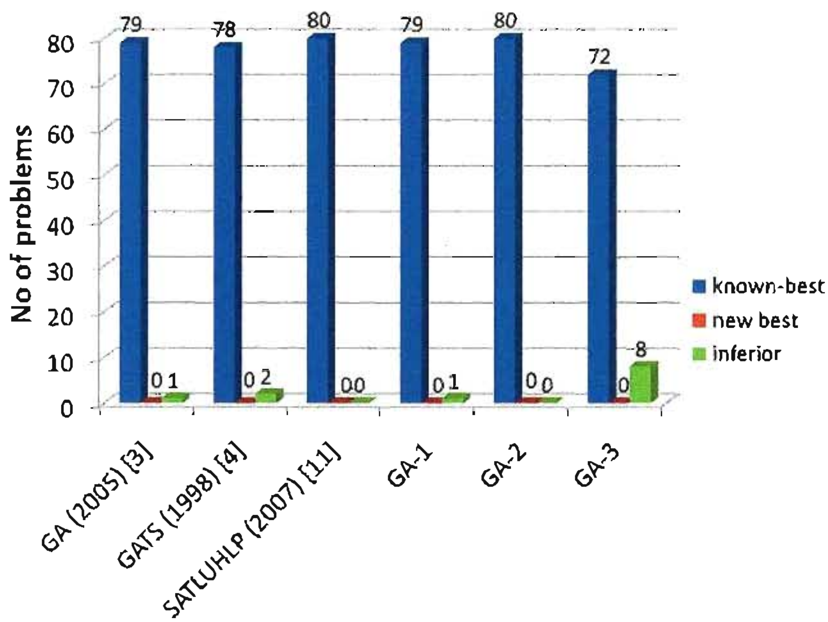


Figure 5.7: USAHLP: Performance in terms of the number of known or new best solutions found, set representation, CAB data

5.4.2 Computational Results Using List-based Representation

A summary of the performance comparison for list representation in terms of problems solved is given in Figure 5.8. As the figure shows, GA-2 was also the most successful on the list representation. It solved all the CAB problems to optimality. Its performance thus matched that of SATLUHLP (2007) [11]. GA-1 gave comparable performance by finding optimal solutions to 79 of the 80 CAB problems. Its performance is thus matching that of GA (2005) [3] and GATS (1998) [4]. GA-3 solved 69 of the 80 CAB problems. For the remaining problems, the quality of its solutions was comparable with that of the optimal solutions. Detailed results are given in tables C.11 through C.14, Appendix C.

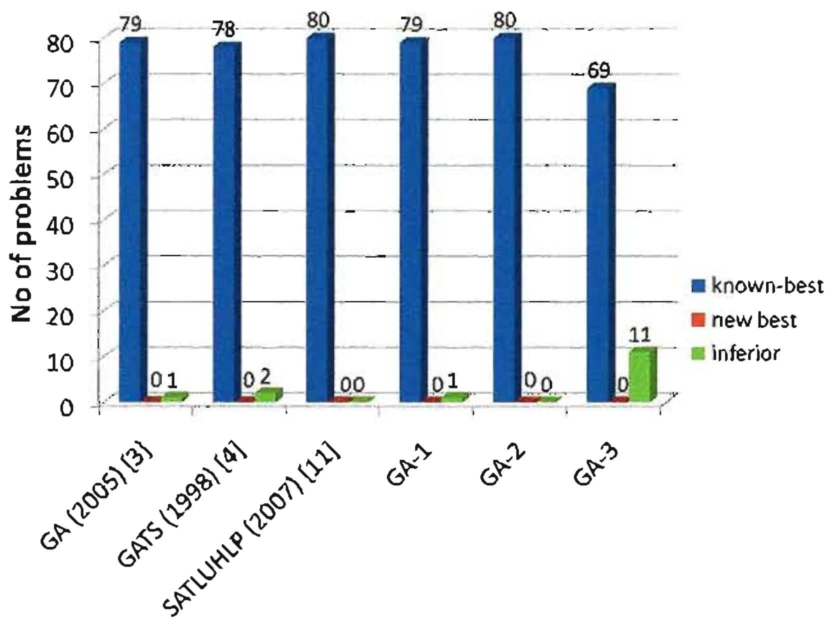


Figure 5.8: USAHLP: Performance in terms of known or new best solutions found, list representation , CAB data

5.4.3 Comparing Performance of the Best GAs with Existing GA Methods

Of all the GAs, GA-2 has the best performance for CAB data on both representations i.e., set and list representations. The chart in Figure 5.9 gives its performance comparison with known GA methods for the uncapacitated SAHLP in terms of the number of problems solved and quality of solutions. As the chart shows, it solved all the CAB problems to optimality compared to 79 problems solved by GA (2005) [3] and GATS (1998) [4] each. Its performance on CAB data is thus better than that of GA (2005) [3] and GATS (1998) [4].

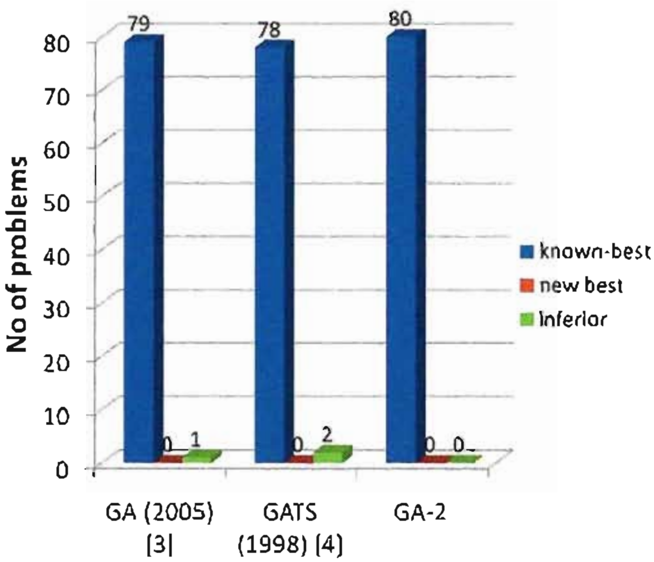


Figure 5.9: USAHLP : Performance comparison of the best GA i.e., GA-2 on CAB problems with existing GA methods

5.5 Conclusions

Tables 5.1 and 5.2 give an overall summary of the GAs performance on the uncapacitated SAHLP problems in terms of number of problems successfully solved. A total of 94 problems were considered, which included 14 problems from AP data set and 80 problems from the CAB data set. As can be seen from the tables, the proposed GAs were also effective on the uncapacitated problems. Performance of GA-1 based on the Double-Cluster Exchange Crossover (DCEC) and GA-2 based on the Multi-Cluster Exchange Crossover (MCEC) was similar on both the representations i.e., Set and List representations, whereas GA-3 showed modest performance relative to GA-1 and GA-2. Comparison with other methods for the uncapacitated SAHLP as given in the preceding sections indicate the effectiveness of the proposed approach for the uncapacitated SAHLP.

Table 5.1: Set representation

	GA-1 (DCEC)	GA-2 (MCEC)	GA-3 (BCRC)
New best	0 / 94	0 / 94	0 / 94
Known-best	91 / 94	92 / 94	79 / 94
Inferior	3 / 94	2 / 94	15 / 94
Overall	91 / 94	92 / 94	79 / 94

Table 5.2: List representation

	GA-1 (DCEC)	GA-2 (MCEC)	GA-3 (BCRC)
New best	0 / 94	0 / 94	0 / 94
Known-best	91 / 94	92 / 94	77 / 94
Inferior	3 / 94	2 / 94	17 / 94
Overall	91 / 94	92 / 94	77 / 94

Chapter 6

Statistical Analysis of the GAs Behaviour

6.1 Test Parameters

To measure significance of difference in the means of the solution populations of the GAs i.e., GA-1, GA-2, and GA-3 on the two representations i.e., set and list representations, two-tailed students t-test for unpaired two-sample data was performed on samples of solutions produced by the algorithms. Following parameters were used in the test.

1. Sample size: 30
2. Degrees of freedom (df) : 58
3. Confidence level: 95
4. Null hypothesis (H_o): The difference between the mean of the solution populations that can be produced by the given pair of GAs is statistically insignificant i.e. $\mu_1 = \mu_2$ where μ_1 is the population mean of the first GA's solutions and μ_2 is the population mean of the second GA's solution.
5. Alterante hypothesis (H_a): The difference between the mean of the solution populations that can be produced by the given pair of algorithm is statistically significant i.e. $\mu_1 \neq \mu_2$

The test was performed for selected AP (Australian Post) problems. Furthermore, two cases were considered; how statistically significant is the difference in the means of the solution population produced by a given pairs of the GAs i.e., GA-1, GA-2, and GA-3, with different crossovers but the same representation i.e., either set or list representation; how significant is the difference in the means of the solution population by a GA with the same crossover but two different representations i.e., the set representation and the list representation.

6.2 Test of Significance for the Representations

The t-test was performed to measure the significance of difference in the mean of the populations produced by the same GA on two different representations. For the capacitated SAHLP, 12 problems with nodes more than or equal to 50 were considered. For the uncapacitated SAHLP, t-test for all the 14 problems in the AP data set was performed. The results for both the capacitated and uncapacitated SAHLP are presented in Figure 6.1.

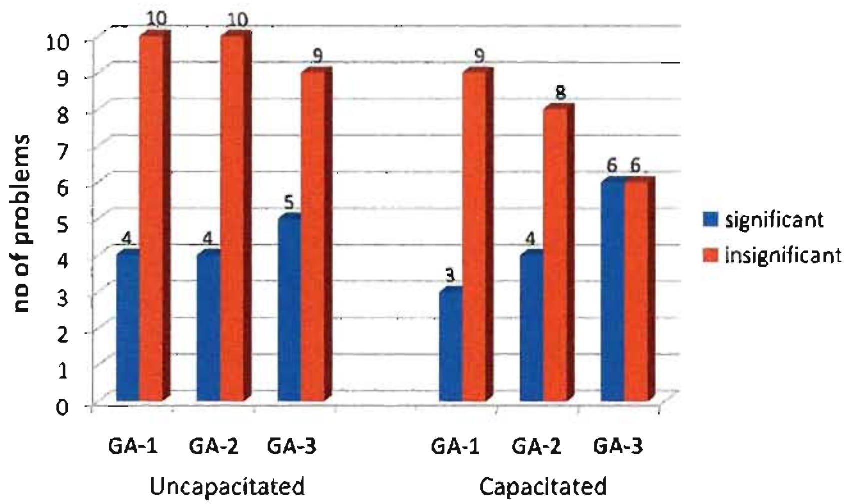


Figure 6.1: SAHLP: t-test for representations

6.2.1 Uncapacitated SAHLP

The result for the GA-1 shows that difference in the means of the solution population produced by the GA-1 on set and list representation was insignificant in the case of 10/14 problems and significant for only one i.e., 4/14 problem. For the GA-2 and GA-3, the insignificant cases are 10/14 and 9/14 respectively. Thus it may be concluded that the statistical behaviour of all the GAs remain the same irrespective of which solution representation i.e., set representation or list representation, is used.

6.2.2 Capacitated SAHLP

For the capacitated SAHLP, the significance in difference of means of the slution population produced by GA-1, GA-2, and GA-3 on the two representation i.e., the set and the list representations, is insignificant for 9/12, 8/12, and 6/12 problems respectively compared to the 3/12, 4/12, and 6/12 problems in which it is significant. This indicates that the statistical behaviur of the three GAs is largely the same regardless of the representation used.

6.3 Test of Significance for the Crossovers

In this section, the results of the t-test for different GA pairs using the same representation are presented. As each of the GA i.e., GA-1 (with Double-Cluster Exchange Crossover), GA-2 (with Multi-Cluster Exchange Crossover), and GA-3 (with Best-Cost Routing Crossover), is based on a different crossover, so the test, in effect, measures the statistical behaviour of different crossovers when the representation is the same. Given below, is a description of the results.

6.3.1 Capacitated SAHLP based Test

The t-test for the capacitated SAHLP was performed for 12 problems with 50 or more nodes. Figure 6.2. gives the results. The label "significant" in the figure denotes the statistical significance and the label "insignificant" denotes the statistical insignificance of the difference in the means of the solution populations of the given GA pairs. The same terms have been used in the remaining figures in this chapter to indicate the significance or insignificance of the difference in the statistical behaviour of the GAs. In the first case i.e., GA-1 and GA-2 on set representation, the difference in the means for 8/12 problems was found to be statistically significant whereas that of the remaining 4/12 was found to be insignificant Thus overall, the difference in behaviour of GA-1 with Double=Cluster Exchange Crossover (DCEC) and GA-2 with Multi-Cluster Exchange Crossover (MCEC) is significant. For the rest of the cases, the difference is overwhelmingly significant as shown by the diagram. Thus, it can be concluded that the GAs have statistically different behaviour for the capacitated SAHLP.

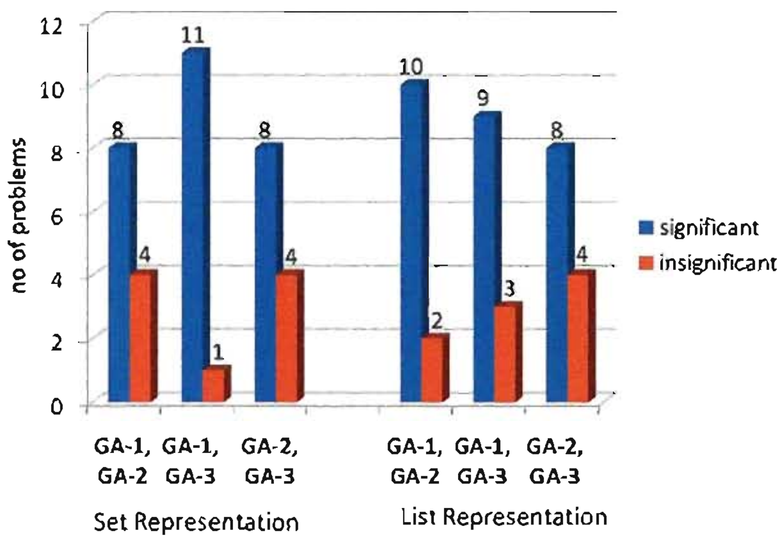


Figure 6.2: CSAHLP: t-test for crossovers, no of problems=12

6.3.2 Uncapacitated SAHLP based Test

The number of problems in the t-test for the uncapacitated SAHLP was 14 (Figure 6.3). For the GA-1 and GA-2 pair, the difference in the means of the solution populations was significant for 9/12 problems indicating that GA-1 and GA-2 have statistically different behaviour on the set representation. On the list representation on the other hand, the significant cases are 10/14. Similarly, for the case GA-1 and GA-3 on set representation, the difference in means of the slution population produced by the two GAs is significant for 11/14 problems and insignificant for the remaining 3/14 problems indicating that GA-1 with Double-Cluster Exchange Crossover (DCEC) and GA-3 with Best Cost Routing Crossover (BCRC) have different behaviour for set representation. The figure shows that the difference between GA-1 and GA3 and GA-2 and GA-3 is significant for larger number of problems.

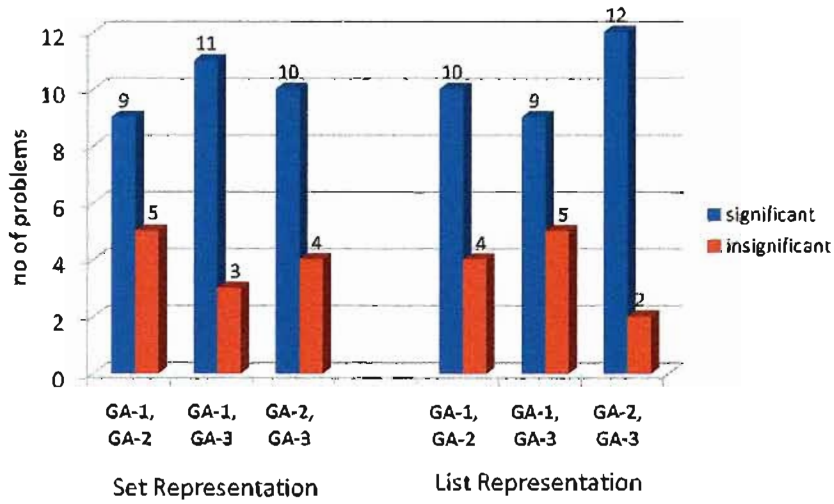


Figure 6.3: USAHLP: t-test for crossovers, no of problems=14

Chapter 7

Conclusions

Most practical combinatorial optimization problems are NP-hard. There are no polynomial algorithms developed and their non-existence is believed. Research on combinatorial optimization based on meta-heuristics is an active current research topic that has gained popularity especially since the 90s. Meta-heuristic approaches seek approximate solutions in polynomial time instead of exact solutions, which would be at intolerably high cost. Although various meta-heuristics have been proposed in the literature for the Single Allocation Hub Location Problem, work using genetic algorithms is limited especially for the capacitated version where only one genetic algorithm approach is currently available. This thesis sort to bridge this gap by proposing an application of genetic algorithms for both the capacitated and incapacitated versions of the SAHLP. An empirical study for the GA based on two different representations and three crossover operators was presented.

Three GAs i.e., GA-1, GA-2, and GA-3, based on the Multi-Cluster Exchange (MCEC), Double-Cluster Exchange (DCEC), and Best-Cost Routing Crossover (BCRC) respectively were designed and run on both the representations i.e. Set and List representations. The effectiveness of the implementations was checked with standard benchmark problems from AP and CAB data sets and the results were presented in chapter 4 and 5.

As can be seen from the results the GA-implementations based on the new crossover approach i.e., GA-1, GA-2, and GA-3 yielded high-quality solutions to both versions of the Single Allocation Hub Location Problem. GA-1 and GA-2 based on the Double-Cluster Exchange and Multiple Cluster Exchange crossovers respectively were able to solve small-sized capacitated AP problems to optimality. Further, they found comparable and in some cases better solutions to the large-sized capacitated SAHLP problems. Furthermore, GA-1 and GA-2 also obtained optimal solutions for the CAB problems and comparably better solutions for the AP problems. The representations i.e., the Set-based and the List-based representations, had significant impact on the performance of the crossovers. The performance of GA-1 and GA-2 was better for set and list representations respectively. However, the Student's t-test indicates that the average performance of each GA was similar for both the representations.

Although, the proposed GA approach produced good results, there is still room for improvement especially in the case of larger SAHLP problems. These observations indicate that the proposed GA approach has good potential for further application and refinement. In future, this research work will be extended in the following directions.

- The proposed GA method will be applied to other hub location problems like Multiple Allocation Hub Location Problem (MAHLP), Single and Multiple Allocation p-Hub Problems, etc., Furthermore, its behaviour will be studied for related problems like Facility Location Problems, Bin Packing Problem, etc.

- The efficiency of the technique handling the hub capacity constraint seems to have a significant impact on the performance of the GA in the case of the capacitated SAHLP. The issue of an efficient technique for handling the capacity constraint will be further explored.

- Although good results were obtained for large problems for the capacitated version of the SAHLP, further work is needed in the design of the GA to better handle larger problem instances for the uncapacitated version.

- A Fitness landscape analysis for the Single Allocation Hub Location Problems should be carried out. Gaining insight into the structure of combinatorial optimization problems by employing search space analysis is an important step in designing efficient genetic operators for the problem. By employing crossover-based and mutation-based landscapes, the efficiency of the genetic operators and representation techniques presented in this thesis can be shown. Furthermore, a search space analysis can help explain what key characteristics make it hard for a certain class of heuristics. Ultimately, the domain-specific knowledge gained from the search space analysis will aid in the design of more effective genetic algorithms (or other meta-heuristics) especially for larger problem instances.

Bibliography

[1] T. Ernst Andreas, Krishanmoorthy Mohan, "Solution Algorithms for the Capacitated Single Allocation Hub Location Problem", Annals of Operations Research 86(1999) 141-159

[2] Randall Marcus, "Solution Approaches for the Capacitated Single Allocation Hub Location Problem Using Ant Colony Optimization", Comput Optim Appl (2008) 39, 239-261

[3] Topcuoglo, Corut, Ermis, Yelmaz, "Solving the Un-capacitated Hub Location Problem Using Genetic Algorithm", Computer and Operations Research 32(2005) 967-984

[4] Abdinnour-Helm Sue, "A Hybrid Heuristic for the Un-capacitated Hub Location Problem", European Journal of Operational Research 106 (1998) 489-499

[5] Katrica, Satnimirovic, Tosic, Filipovic, "Two Genetic Algorithms for Solving the Un-capacitated Single Allocation p-Hub Median Problem", European Journal of Operations Research 182(2007) 15-28

[6] Hordjik Wim, "A Measure of Landscapes", wim@santafe.edu

[7] http://people.brunel.ac.uk/ mastjjb/jeb/orlib/phubinfo.html

[8] Jones Terry, "Genetic Algorithms, Fitness Landscape, and Search", Ph. D. Thesis

[9] Stadler Peter, "Fitness Landscapes", Austria

[10] Almur Sibel, Kara Bahar, "Network Hub Location Problem: State of the Art", European Journal of Operational Research 190 (2008) 121

[11] Chen, J.F., "A hybrid heuristic for the uncapacitated single allocation hub location problem", Omega 35, 211220.

[12] Cunha, C.B., Silva, M.R., 2007. A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. European Journal of Operational Research 179, 747758.

[13] Satnimirovic Z., "Solving the Capacitated Single Allocation Hub Location Problem Using Genetic Algorithm",

[14] Aykin, T., Networking policies for hub-and-spoke systems with application to the air transportation system, Transportation Science, 29, 201-221 (1995).

[15] Bania, N., P. Bauer and T. Zlatoper, U.S. air passenger service: a taxonomy of route networks, hub locations, and competition, Logistics and Transportation Review, 34, 53-74 (1998).

[16] Skorin-Kapov, D. and J. Skorin-Kapov, On tabu search for the location of interacting hub facilities, European Journal of Operational Research, 73, 502-509 (1994).

[17] Kuby, M. and R. Gray, Hub network design problem with stopovers and feeders: case of Federal Express, Transportation Research, 27, 1-12 (1993).

[18] Klincewicz, J., Hub location in backbone tributary network design: a review, Location Science, 6, 307-335 (1998).

[19] Lee, Y., B. Lim and J. Park, A hub location problem in designing digital data service networks: Lagrangian relaxation approach, Location Science, 4, 185-194 (1996).

[20] Abdinnour-Helm, Sue, Using simulated annealing to solve the p-hub median problem, International Journal of Physical Distribution and Logistics Management, 31, 203-220 (2001).

[21] Don, T., S. Harit, J. English and G. Whicker, Hub and spoke networks in truckload trucking: configuration, testing, and operational concerns, Logistics and Transportation, 31, 209-237 (1995).

[22] Ronald S., "Robust Encoding in Genetic Algorithms: A Survey of Encoding Issues", IEEE International Conference on Evolutionary Computation, 43-48 (1997)

[23] Ombuki-Berman B. M., Runka A., Hanshar F. T., "Waste Collection Vehicle Routing Problem with Time Windows using Multi-Objective Genetic Algorithms", From Proceeding (574) Computational Intelligence, 2007

[24] O'Kelley M. E., "A Qudratic Integer Program for the Location of Interacting Hubs", European Journal of Operational Research 32(1987) 393-404

[25] Contreras I., Diaz A. J., Fernandez E., "Lagrangian relaxation for the capacitated hub location problem with single assignment", OR Spectrum (2009) 31:483505

[26] D.E. Goldberg, R. Lingle. "Alleles, loci, and the trave-ling salesman problem." In J.J. Grefenstette (ed.), Pro-ceedings of the First International Conference on Genetic Algorithms and their Applications, Lowrence Erlbaum, Hillsdale, 1985, 154-159.

[27] Beasley D., Bull D. R., Martin R. R., "An overview of Genetic Algorithms", University Computing, 15(4), 170-181, UCSIA

[28] Goldberg D., Deb K., "A comparative analysis of selection schemes used in genetic algorithms". Foundations of Genetic Algorithms, Morgan Kaufmann Publishers

[29] Goldberg D. E., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley(Reading Mass), 1989

[30] Reeves C., "Hybrid Genetic Algorithms for Binpacking and Related Problems", Annals of Operation Research, 63, 371-396 (1993)

[31] Namada T., Nakano R., "Genetic Algorithms for Job-Shop Scheduling Problems", In Proceedings of Modern Heuristic for Decision Support, Morgan Kaufmann, 474-479 (1997)

[32] Fang-Chih Tien, Kuang-Han Hsieh, Chun-Yuan Cheng and Chi-Shuan Liu, "Using Hybrid Genetic Algorithms to Solve Discrete Location Allocation Problems with Rectilinear Distance," Journal of the Chinese Institute of Industrial Engineers, 24, 1-19 (2007).

[33] Genetic algorithms + data structures = evolution programs / Zbigniew Michalewicz. 3rd Edition

[34] Spears W., "Adapting Crossover in a Genetic Algorithm", Naval Research Laboratory AI Center Report AIC-92-025. Washington, 1995

[35] An introduction to genetic algorithms / Melanie Mitchell. Cambridge, Mass. : MIT Press, 1996.

[36] Hinterding R., Gielewski H. and Peachey T. C., "The Nature of Mutation in Genetic Algorithms", In Proceedings of the Sixth International Conference on Genetic Algorithms, edited by L.J. Eshelman, Morgan Kaufmann, 65–72 (1995)

[37] David M. Tate and Alice E. Smith, "Expected Allele Coverage and the Role of Mutation in Genetic Algorithms", Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, 31–37 (1993)

[38] Oliver I. M., Smith D. J., Holland J. R. C., "A Stufy of Permutation Crossover Operators on the Traveling Salesman Problem", International Conference on Genetic Algorithms and their Applications, MIT, 224-230 (1997)

[39] "A Problem-Specific Genetic Algorithm for Path Planning of Mobile Robot in Greenhouse", Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management, Springer Boston, 207/2006, 211-216 (2006)

[40] Z. Drezner,"A new genetic algorithm for the quadratic assignment problem", INFORMS Journal on Computing, 15(2002), pp.320-330.

[41] Radcliffe, N. J., Forma Analysis and Random Respectful Recombination. Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann. 1991.

[42] Ombuki B., Ross B. J, Hansher F., Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows, Applied Intelligence 24, 17-30, 2006, Springer Science.

[43] Saloman Ralph, The Influence of Different Encoding Schemes on the Computational Complexity of Genetic Algorithms in Function Optimization", Proceedingds of The Fourth International Conference on Parallel Problem Solving from Nature (PPSN IV) pp. 227-235, Springer-Verlog, Berlin

Appendix A

Results for the Crossover Rate of 0.8 and Mutation Rate of 0.2

Table A.1: CSAHLP: AP Data, Crossover = 0.8, Mutation = 0.2

Problem	Cost	GA-1 Set	GA-1 List	GA-2 Set	GA-2 List	GA-3 Set	GA-3 List
10LL	224250.05	√	√	√	√	224706.43	224706.44
10LT	250992.26	√	√	254112.39	√	√	√
10TL	263399.94	√	√	√	√	263763.43	263763.43
10TT	263399.94	√	√	√	√	√	√
20LL	234690.94	√	√	√	√	√	√
20LT	253517.40	√	√	√	√	√	√
20TL	271128.18	√	√	√	√	√	√
20TT	296035.40	√	298441.12	298547.09	√	√	√
25LL	238977.95	√	√	√	√	√	√
25LT	276372.50	√	√	√	√	√	√
25TL	310317.64	√	√	√	√	√	√
25TT	348369.15	√	√	√	√	√	√
40LL	241955.71	√	√	√	√	√	√
40LT	272218.32	272455.80	274191.88	√	√	271142.54	√
40TL	298919.01	√	√	√	√	√	√
40TT	354874.10	357441.31	359121.07	356873.33	√	√	√
50LL	238520.59	√	√	√	√	√	√
50LT	272897.49	273159.66	273159.66	273002.02	√	274142.54	273155.43
50TL	319015.77	319375.12	320176.53	√	√	319375.23	319375.23
50TT	417440.99	423886.89	422671.56	422657.74	422287.75	422345.53	423751.15
100LL	246713.97	246755.13	246910.54	246755.13	246729.32	246910.37	246755.13
100LT	256250.41	256987.63	258331.34	257131.31	256784.45	257344.12	257101.12
100TL	362950.09	364820.15	365401.31	365401.53	364572.12	365333.56	365401.53
100TT	474680.32	480130.65	481667.45	481334.12	479754.23	482334.75	483432.61
200LL	241992.97	242910.77	243334.43	242337.22	241997.83	243117.56	242455.76
200LT	268894.41	271854.34	270334.17	272055.72	269621.45	273155.97	271331.43
200TL	273443.81	273660.19	273785.45	273538.74	273550.17	273687.12	273602.45
200TT	292754.97	293241.75	295755.42	294475.19	292245.53	295755.42	294141.66

Table A.2: USAHLP: AP Data, Crossover = 0.8, Mutation = 0.2

		GA-1		GA-2		GA-3	
Problem	Cost	Set	List	Set	List	Set	List
10L	224249.82	√	√	√	√	√	224706.29
10T	263402.13	√	√	√	√	263763.43	263763.43
20LL	234690.11	√	√	√	√	√	√
20LT	271128.41	√	√	√	√	√	√
25LL	236649.69	√	√	√	√	236797.68	√
25LT	295670.39	√	√	√	√	√	√
40LL	240985.51	√	√	√	√	241260.20	241260.20
40LT	293163.38	√	274191.88	√	√	√	√
50LL	237420.69	√	√	238520.58	√	237518.79	237518.79
50LT	300420.87	√	√	√	√	√	√
100LL	238017.53	238199.12	238199.12	√	√	238492.88	238383.20
100LT	305101.07	√	√	√	√	√	√
200LL	228044.77	234109.34	234109.54	234185.89	234052.81	238635.51	238963.80
200LT	233537.93	272212.8	272982.14	279848.29	272205.32	272757.80	276874.22

Table A.3: USAHLP, CAB data, nodes = 10, Crossover = 0.8, Mutation 0.2

α	f	Optimal	Hubs	GA-1 Set	GA-1 List	GA-2 Set	GA-2 List	GA-3 Set	GA-3 List
0.2	100	791.93	4,6,7	√	√	√	√	√	√
	150	915.99	7,9	√	√	√	√	√	√
	200	1015.99	7,9	√	√	√	√	√	√
	250	1115.99	7,9	√	√	√	√	√	√
0.4	100	867.91	4,6,7	√	√	√	√	√	√
	150	974.30	7,9	√	√	√	√	√	√
	200	1074.30	7,9	√	√	√	√	√	√
	250	1174.30	7,9	√	√	√	√	√	√
0.6	100	932.62	7,9	√	√	√	√	√	√
	150	1032.62	7,9	√	√	√	√	√	√
	200	1131.05	9	√	√	√	√	√	√
	250	1181.05	9	√	√	√	√	√	√
0.8	100	999.94	7,9	√	√	√	√	√	√
	150	1081.05	4	√	√	√	√	√	√
	200	1131.05	4	√	√	√	√	√	√
	250	1181.05	4	√	√	√	√	√	√
1.0	100	1031.04	4	√	√	√	√	√	√
	150	1081.05	4	√	√	√	√	√	√
	200	1131.05	4	√	√	√	√	√	√
	250	1181.05	4	√	√	√	√	√	√

Table A.4: USAHLP, CAB problems, nodes = 15, Crossover = 0.8, Mutation 0.2

α	f	Optimal	Hubs	GA-1 Set	GA-1 List	GA-2 Set	GA-2 List	GA-3 Set	GA-3 List
0.2	100	1030.07	3,4,7,12,14	1094.9	√	√	√	1049.9	1032.60
	150	1239.77	4,7,12,14	√	√	√	√	√	1249.97
	200	1381.28	4,12	√	√	√	√	√	√
	250	1481.28	4,12	√	√	√	√	√	√
0.4	100	1179.71	4,7,12,14	√	√	√	√	√	1182.70
	150	1355.09	4,7,12	√	√	√	√	√	1358.30
	200	1462.62	4,12	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√
0.6	100	1309.92	4,7,12	1330.33	1330.33	1330.33	√	1330.33	1330.33
	150	1443.97	4,12	1456.66	1456.66	1456.66	√	1456.66	1456.66
	200	1506.66	4	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√
0.8	100	1390.06	4,11	√	√	√	√	√	√
	150	1456.66	4	√	√	√	√	√	√
	200	1506.66	4	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√
1.0	100	1406.66	4	√	√	√	√	√	√
	150	1456.66	4	√	√	√	√	√	√
	200	1506.66	4	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√

Table A.5: USAHLP, CAB problems, nodes = 20, Crossover = 0.8, Mutation 0.2

α	f	Optimal	Hubs	GA-1 Set	GA-1 List	GA-2 Set	GA-2 List	GA-3 Set	GA-3 List
0.2	100	***	4,7,12,14,17	967.74	967.74	967.74	967.74	967.74	977.21
	150	1174.53	4,12,17	√	√	√	√	√	√
	200	1324.53	4,12,17	√	√	√	√	√	√
	250	1474.53	4,12,17	√	√	√	√	√	√
0.4	100	1127.09	1,4,12,17	1136.65	√	1136.65	√	1136.65	1136.65
	150	1297.76	4,12,17	√	√	√	√	√	√
	200	1442.56	4,17	√	√	√	√	√	√
	250	1542.56	4,17	√	√	√	√	√	√
0.6	100	1269.15	1,4,12,17	1270.99	1270.99	1270.99	√	1270.99	1270.99
	150	1406.04	4,17	√	√	√	√	√	√
	200	1506.04	4,17	√	√	√	√	√	√
	250	1570.91	6	√	√	√	√	√	√
0.8	100	1369.52	4,17	√	√	√	√	√	√
	150	1469.52	4,17	√	√	√	√	√	√
	200	1520.91	6	√	√	√	√	√	√
	250	1570.91	6	√	√	√	√	√	√
1.0	100	1410.07	4,20	√	√	√	√	√	√
	150	1470.91	6	√	√	√	√	√	√
	200	1520.91	6	√	√	√	√	√	√
	250	1570.91	6	√	√	√	√	√	√

Table A.6: USAHLP, CAB problems, nodes = 25, set representation

α	f	Optimal	Hubs	GA-1 Set	GA-1 List	GA-2 Set	GA-2 List	GA-3 Set	GA-3 List
0.2	100	1029.63	4,12,17,24	√	√	√	√	√	√
	150	1217.34	4,12,17	√	√	√	√	√	√
	200	1367.34	4,12,17	√	√	√	√	√	√
	250	1500.90	12,20	√	√	√	√	√	√
0.4	100	1187.51	1,4,12,17	1194.49	√	1294.49	√	1200.34	1194.49
	150	1351.69	4,12,18	√	√	√	√	√	√
	200	1501.62	12,20	√	√	√	√	√	√
	250	1601.62	12,20	√	√	√	√	√	√
0.6	100	1333.56	2,4,12	√	√	√	√	√	1333.99
	150	1483.56	2,4,12	√	√	√	√	√	√
	200	1601.20	12,20	√	√	√	√	√	√
	250	1701.20	12,20	√	√	√	√	√	√
0.8	100	1458.83	24,3,11	√	√	√	√	√	1459.74
	150	1594.08	12,20	√	√	√	√	√	√
	200	1690.57	5	√	√	√	√	√	√
	250	1740.57	5	√	√	√	√	√	√
1.0	100	1556.63	7,19	1562.15	1559.19	√	1559.19	√	1559.19
	150	1640.57	5	√	√	√	√	√	√
	200	1690.57	5	√	√	√	√	√	√
	250	1740.57	5	√	√	√	√	√	√

Appendix B

Detailed Results for the Capacitated Single Allocation Hub Location Problem

Table B.1: CSAHLP: Comparison with current GA approaches, set-representation, AP Data

Problem	Known-best Ernst[1]	GA(2008) Stanimirovic[13]	GA-1	GA-2	GA-3
10LL	224250.05	√	√	√	224706.29
10LT	250992.26	√	√	√	√
10TL	263399.94	√	√	√	√
10TT	263399.94	√	√	√	√
20LL	234690.94	√	√	√	√
20LT	253517.40	√	√	√	√
20TL	271128.18	√	√	√	√
20TT	296035.40	√	√	√	√
25LL	238977.95	√	√	√	√
25LT	276372.50	√	√	√	√
25TL	310317.64	√	√	√	310493.20
25TT	348369.15	√	√	√	√
40LL	241955.71	√	√	√	√
40LT	272218.32	√	√	√	272455.80
40TL	298919.01	√	√	√	√
40TT	354874.10	356507.86	√	√	√
50LL	238520.59	√	√	√	√
50LT	272897.49	√	√	√	√
50TL	319015.77	√	√	√	√
50TT	417440.99	422794.56	√	√	418269.90
100LL	2246713.97	√	√	√	246755.13
100LT	256639.38	256207.52	256250.32	**256183.42**	**256183.43**
100TL	362950.09	364515.24	√	√	365247.39
100TT	474670.32	475156.75	**474660.51**	474667.32	478937.94
200LL	241992.97	√	√	√	241993.97
200LT	268894.41	270202.25	**268661.14**	269494.09	272089.92
200TL	273443.81	√	√	√	273502.88
200TT	292734.97	**291830.66**	291969.46	291973.07	292154.47

Table B.2: CSAHLP: Comparison with current GA approaches, list-representation, AP Data

Problem	Best knownl value Ernst[1]	GA(2008) Stanimirovic[13]	GA-1	GA-2	GA-3
10LL	224250.05	√	√	√	224706.29
10LT	250992.26	√	√	√	√
10TL	263399.94	√	√	√	√
10TT	263399.94	√	√	√	√
20LL	234690.94	√	√	√	√
20LT	253517.40	√	√	√	√
20TL	271128.18	√	√	√	√
20TT	296035.40	√	√	√	√
25LL	238977.95	√	√	√	√
25LT	276372.50	√	√	√	√
25TL	310317.64	√	√	√	310493
25TT	348369.15	√	√	√	√
40LL	241955.71	√	√	√	√
40LT	272218.32	√	√	√	√
40TL	298919.01	√	√	√	√
40TT	354874.10	356507.86	√	√	√
50LL	238520.59	√	√	√	√
50LT	272897.49	√	√	√	√
50TL	319015.77	√	√	√	319784
50TT	417440.99	422794.56	√	√	418215.16
100LL	2246713.97	√	√	√	246755.13
100LT	256639.38	256207.52	256250.41	**256155.33**	256455.24
100TL	362950.09	364515.24	√	√	364515.45
100TT	474670.32	475156.75	474287.49	**474184.94**	476568.42
200LL	241992.97	√	√	√	241992.97
200LT	268894.41	270202.25	268661.14	**267827.97**	269276.34
200TL	273443.81	√	273541.82	√	273502.88
200TT	292734.97	**291830.66**	292237.69	291891.69	292258.72

Table B.3: CSAHLP: Comparison with non-GA approaches, set-representation, AP Data

Problem	Best known value Ernst[1]	SA-RDH(1999) Ernst[1]	GA-1	GA-2	GA-3
10LL	224250.05	√	√	√	224706.29
10LT	250992.26	√	√	√	√
10TL	263399.94	√	√	√	√
10TT	263399.94	√	√	√	√
20LL	234690.94	√	√	√	√
20LT	253517.40	√	√	√	√
20TL	271128.18	√	√	√	√
20TT	296035.40	√	√	√	√
25LL	238977.95	√	√	√	√
25LT	276372.50	√	√	√	√
25TL	310317.64	√	√	√	310493.20
25TT	348369.15	√	√	√	√
40LL	241955.71	√	√	√	√
40LT	272218.32	√	√	√	272455.80
40TL	298919.01	√	√	√	√
40TT	354874.10	√	√	√	√
50LL	238520.59	√	√	√	√
50LT	272897.49	√	√	√	√
50TL	319015.77	√	√	√	√
50TT	417440.99	√	√	√	418269.90
100LL	246713.97	√	√	√	246755.13
100LT	256639.38	√	256250.32	**256183.42**	**256183.43**
100TL	362950.09	√	√	√	365247.39
100TT	474670.32	√	**474660.51**	474667.32	478937.94
200LL	241992.97	√	√	√	241993.97
200LT	268894.41	√	**268661.14**	269494.09	272089.92
200TL	273443.81	√	√	√	273502.88
200TT	292734.97	√	291969.46	**291973.07**	292154.47

Table B.4: CSAHLP: Comparison with non-GA approaches, list-representation, AP Data

Problem	Best known value Ernst[1]	SA-RDH(1999) Ernst[1]	GA-1	GA-2	GA-3
10LL	224250.05	√	√	√	224706.29
10LT	250992.26	√	√	√	√
10TL	263399.94	√	√	√	√
10TT	263399.94	√	√	√	√
20LL	234690.94	√	√	√	√
20LT	253517.40	√	√	√	√
20TL	271128.18	√	√	√	√
20TT	296035.40	√	√	√	√
25LL	238977.95	√	√	√	√
25LT	276372.50	√	√	√	√
25TL	310317.64	√	√	√	310493.26
25TT	348369.15	√	√	√	√
40LL	241955.71	√	√	√	√
40LT	272218.32	√	√	√	√
40TL	298919.01	√	√	√	√
40TT	354874.10	√	√	√	√
50LL	238520.59	√	√	√	√
50LT	272897.49	√	√	√	√
50TL	319015.77	√	√	√	319784.37
50TT	417440.99	√	√	√	418215.16
100LL	2246713.97	√	√	√	246755.13
100LT	256639.38	√	256250.41	**256155.33**	256455.24
100TL	362950.09	√	√	√	364515.45
100TT	474670.32	√	474287.94	**474184.94**	476568.42
200LL	241992.97	√	√	√	241992.97
200LT	268894.41	√	268661.14	**267827.97**	269276.34
200TL	273443.81	√	273541.82	√	273502.88
200TT	292734.97	√	292237.69	291891.69	292258.72

Appendix C

Detailed Results for the Uncapacitated Single Allocation Hub Location Problem

Table C.1: USAHLP: Comparison with current GA methods, set-representation, AP data

Problem	GA(2005) Topcouglo[3]	GA(1998) Abdinour-Helm[4]	GA-1	GA-2	GA-3
10L	224249.82	√	√	√	√
20L	234690.11	√	√	√	√
25L	236649.69	√	√	√	236797.68
40L	240985.51	√	√	√	241260.24
50L	237420.69	300226.47	√	√	237518.80
100L	238017.53	695705.82	**238016.27**	**238016.27**	238492.88
200L	228944.77	1967625.73	233802.45	233425.75	234109.34
10T	263402.13	√	√	√	√
20T	271128.41	√	√	√	√
25T	295670.39	√	√	√	√
40T	293163.38	345386.77	√	√	√
50T	300420.87	411145.42	√	√	√
100T	305101.07	4369213.98	**305096.76**	**305096.76**	**305096.76**
200T	233537.93	11911942.30	272516.86	272424.34	272842.91

Table C.2: USAHLP: Comparison with current GA methods. list-representation, AP data

Problem	GA(2005) Topcouglo[3]	GATS(1998) Abdinour-Helm[4]	GA-1	GA-2	GA-3
10L	224249.82	√	√	√	224706.28
20L	234690.11	√	√	√	√
25L	236649.69	√	√	√	√
40L	240985.51	√	√	√	√
50L	237420.69	300226.47	√	√	237518.51
100L	238017.53	695705.82	**238016.27**	**238016.27**	**238016.27**
200L	228944.77	1967625.73	233802.45	233802.97	234337.21
10T	263402.13	√	√	√	263763.43
20T	271128.41	√	√	√	√
25T	295670.39	√	√	√	√
40T	293163.38	345386.77	√	√	√
50T	300420.87	411145.42	√	√	√
100T	305101.07	4369213.98	**305096.76**	**305096.76**	**305096.76**
200T	233537.93	11911942.30	272168.31	272222.93	273006.21

Table C.3: USAHLP: Comparison with non-GA methods, set-representation, AP data

Problem	SATLUHLP(2007) Chen [11]	SA(2005) Topcouglo[3]	GA-1	GA-2	GA-3
10L	224249.82	√	√	√	√
20L	234690.11	√	√	√	√
25L	236649.69	√	√	√	236797.68
40L	240985.51	√	√	√	241260.24
50L	237420.69	√	√	√	237518.80
100L	238016.27	√	√	√	238492.88
200L	228944.77	√	233802.45	233425.75	234109.34
10T	263402.13	√	√	√	√
20T	271128.41	√	√	√	√
25T	295670.39	√	√	√	√
40T	293163.38	√	√	√	√
50T	300420.87	√	√	√	√
100T	305096.76	305101.07	√	√	√
200T	233537.93	√	272516.86	272424.34	272842.91

Table C.4: USAHLP: Comparison with non-GA approaches, list-representation, AP data

Problem	SATLUHLP(2007) Chen [11]	SA(2005) Topcouglo [3]	GA-1	GA-2	GA-3
10L	224249.82	√	√	√	224706.28
20L	234690.11	√	√	√	√
25L	236649.69	√	√	√	√
40L	240985.51	√	√	√	√
50L	237420.69	√	√	√	237518.51
100L	238016.27	√	√	√	238016.27
200L	228944.77	√	233802.45	233802.97	234337.21
10T	263402.13	√	√	√	263763.43
20T	271128.41	√	√	√	√
25T	295670.39	√	√	√	√
40T	293163.38	√	√	√	√
50T	300420.87	√	√	√	√
100T	305096.76	305101.07	√	√	√
200T	233537.93	√	272168.31	272222.93	273006.21

Table C.5: USAHLP: Performance comparison of GA-1, GA-2, and GA-3 using set-representation, AP data

Problem	known-best[11][4][3]	GA-1	GA-2	GA-3
10L	224249.82	√	√	√
20L	234690.11	√	√	√
25L	236649.69	√	√	236797.68
40L	240985.51	√	√	241260.24
50L	237420.69	√	√	237518.80
100L	238016.27	√	√	238492.88
200L	228944.77	233802.45	233425.75	234109.34
10T	263402.13	√	√	√
20T	271128.41	√	√	√
25T	295670.39	√	√	√
40T	293163.38	√	√	√
50T	300420.87	√	√	√
100T	305096.76 [11]	√	√	√
200T	233537.93	272516.86	272424.34	272842.91
Total best solutions		12/14	12/14	7/14

Table C.6: USAHLP: Performance comparison of GA-1, GA-2, and GA-3 using list-representation, AP data

Problem	known-best[11][4][3]	GA-1	GA-2	GA-3	
10L	224249.82		√	√	224706.28
20L	234690.11	√	√	√	
25L	236649.69	√	√	√	
40L	240985.51	√	√	√	
50L	237420.69	√	√	237518.51	
100L	238016.27	√	√	238515.92	
200L	228944.77	233802.45	233802.97	234337.21	
10T	263402.13	√	√	263763.43	
20T	271128.41	√	√	√	
25T	295670.39	√	√	√	
40T	293163.38	√	√	√	
50T	300420.87	√	√	√	
100T	305096.76 [11]	√	√	√	
200T	233537.93	272168.31	272222.93	273006.21	
Total best solutions		12/14	12/14	7/14	

Table C.7: USAHLP: Comparison with other approaches, set representation, nodes = 10, CAB data

α	f	Optimal Cost	Hubs	GA Topcouglo [3] (2005)	GATS Abdnour[4] (1998)	SATLUHLP Chen. J.[11] (2007)	GA-1	GA-2	GA-3
0.2	100	791.93	4,6,7	√	√	√	√	√	√
	150	915.99	7,9	√	√	√	√	√	√
	200	1015.99	7,9	√	√	√	√	√	√
	250	1115.99	7,9	√	√	√	√	√	√
0.4	100	867.91	4,6,7	√	√	√	√	√	√
	150	974.30	7,9	√	√	√	√	√	√
	200	1074.30	7,9	√	√	√	√	√	√
	250	1174.30	7,9	√	√	√	√	√	√
0.6	100	932.62	7,9	√	√	√	√	√	√
	150	1032.62	7,9	√	√	√	√	√	√
	200	1131.05	9	√	√	√	√	√	√
	250	1181.05	9	√	√	√	√	√	√
0.8	100	999.94	7,9	√	√	√	√	√	√
	150	1081.05	4	√	√	√	√	√	√
	200	1131.05	4	√	√	√	√	√	√
	250	1181.05	4	√	√	√	√	√	√
1.0	100	1031.04	4	√	√	√	√	√	√
	150	1081.05	4	√	√	√	√	√	√
	200	1131.05	4	√	√	√	√	√	√
	250	1181.05	4	√	√	√	√	√	√

Table C.8: USAHLP: Comparison with other approaches, set representation, nodes = 15, CAB data

α	f	Optimal Cost	Hubs	GA Topcouglo [3] (2005)	GATS Abdnour[4] (1998)	SATLUHLP Chen. J.[11] (2007)	GA-1	GA-2	GA-3
0.2	100	1030.07	3,4,7,12,14	√	√	√	√	√	1032.60
	150	1239.77	4,7,12,14	√	√	√	√	√	1249.97
	200	1381.28	4,12	√	√	√	√	√	√
	250	1481.28	4,12	√	√	√	√	√	√
0.4	100	1179.71	4,7,12,14	√	√	√	√	√	1182.70
	150	1355.09	4,7,12	√	√	√	√	√	1358.30
	200	1462.62	4,12	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√
0.6	100	1309.92	4,7,12	√	√	√	√	√	√
	150	1443.97	4,12	√	√	√	√	√	√
	200	1506.66	4	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√
0.8	100	1390.06	4,11	√	√	√	√	√	√
	150	1456.66	4	√	√	√	√	√	√
	200	1506.66	4	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√
1.0	100	1406.66	4	√	√	√	√	√	√
	150	1456.66	4	√	√	√	√	√	√
	200	1506.66	4	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√

Table C.9: USAHLP: Comparison with other approaches, set representation, nodes = 20, CAB data

α	f	Optimal Cost	Hubs	GA Topcouglo [3] (2005)	GATS Abdnour[4] (1998)	SATLUHLP Chen. J.[11] (2007)	GA-1	GA-2	GA-3
0.2	100	***	4,7,12,14,17	967.74	967.74	967.74	967.74	967.74	977.21
	150	1174.53	4,12,17	√	√	√	√	√	√
	200	1324.53	4,12,17	√	√	√	√	√	√
	250	1474.53	4,12,17	√	√	√	√	√	√
0.4	100	1127.09	1,4,12,17	√	√	√	√	√	√
	150	1297.76	4,12,17	√	√	√	√	√	√
	200	1442.56	4,17	√	√	√	√	√	√
	250	1542.56	4,17	√	√	√	√	√	√
0.6	100	1269.15	1,4,12,17	√	√	√	√	√	1270.99
	150	1406.04	4,17	√	√	√	√	√	√
	200	1506.04	4,17	√	√	√	√	√	√
	250	1570.91	6	√	√	√	√	√	√
0.8	100	1369.52	4,17	√	√	√	√	√	√
	150	1469.52	4,17	√	√	√	√	√	√
	200	1520.91	6	√	√	√	√	√	√
	250	1570.91	6	√	√	√	√	√	√
1.0	100	1410.07	4,20	√	√	√	√	√	√
	150	1470.91	6	√	√	√	√	√	√
	200	1520.91	6	√	√	√	√	√	√
	250	1570.91	6	√	√	√	√	√	√

Table C.10: USAHLP: Comparison with other approaches, set representation, nodes = 25, CAB data

α	f	Optimal Cost	Hubs	GA Topcouglo [3] (2005)	GATS Abdnour[4] (1998)	SATLUHLP Chen. J.[11] (2007)	GA-1	GA-2	GA-3
0.2	100	1029.63	4,12,17,24	√	√	√	√	√	√
	150	1217.34	4,12,17	√	√	√	√	√	√
	200	1367.34	4,12,17	√	√	√	√	√	√
	250	1500.90	12,20	√	√	√	√	√	√
0.4	100	1187.51	1,4,12,17	√	√	√	√	√	√
	150	1351.69	4,12,18	√	√	√	√	√	√
	200	1501.62	12,20	√	√	√	√	√	√
	250	1601.62	12,20	√	√	√	√	√	√
0.6	100	1333.56	2,4,12	√	√	√	√	√	1333.99
	150	1483.56	2,4,12	√	√	√	√	√	√
	200	1601.20	12,20	√	√	√	√	√	√
	250	1701.20	12,20	√	√	√	√	√	√
0.8	100	1458.83	2,4,12	√	√	√	√	√	√
	150	1594.08	12,20	√	√	√	√	√	√
	200	1690.57	5	√	√	√	√	√	√
	250	1740.57	5	√	√	√	√	√	√
1.0	100	1556.63	7,19	1559.19	1562.15	√	1559.19	√	1559.19
	150	1640.57	5	√	√	√	√	√	√
	200	1690.57	5	√	√	√	√	√	√
	250	1740.57	5	√	√	√	√	√	√

Table C.11: USAHLP: Comparison with other approaches, list representation, nodes = 10, CAB data

α	f	Optimal Cost	Hubs	GA Topcouglo [3] (2005)	GATS Abdnour[4] (1998)	SATLUHLP Chen. J.[11] (2007)	GA-1	GA-2	GA-3
0.2	100	791.93	4,6,7	√	√	√	√	√	√
	150	915.99	7,9	√	√	√	√	√	√
	200	1015.99	7,9	√	√	√	√	√	√
	250	1115.99	7,9	√	√	√	√	√	√
0.4	100	867.91	4,6,7	√	√	√	√	√	√
	150	974.30	7,9	√	√	√	√	√	√
	200	1074.30	7,9	√	√	√	√	√	√
	250	1174.30	7,9	√	√	√	√	√	√
0.6	100	932.62	7,9	√	√	√	√	√	√
	150	1032.62	7,9	√	√	√	√	√	√
	200	1131.05	9	√	√	√	√	√	√
	250	1181.05	9	√	√	√	√	√	√
0.8	100	999.94	7,9	√	√	√	√	√	√
	150	1081.05	4	√	√	√	√	√	√
	200	1131.05	4	√	√	√	√	√	√
	250	1181.05	4	√	√	√	√	√	√
1.0	100	1031.04	4	√	√	√	√	√	√
	150	1081.05	4	√	√	√	√	√	√
	200	1131.05	4	√	√	√	√	√	√
	250	1181.05	4	√	√	√	√	√	√

Table C.12: USAHLP: Comparison with other approaches, list representation, nodes = 15, CAB data

α	f	Optimal Cost	Hubs	GA Topcouglo [3] (2005)	GATS Abdnour[4] (1998)	SATLUHLP Chen. J.[11] (2007)	GA-1	GA-2	GA-3
0.2	100	1030.07	3,4,7,12,14	√	√	√	√	√	1032.60
	150	1239.77	4,7,12,14	√	√	√	√	√	1249.97
	200	1381.28	4,12	√	√	√	√	√	√
	250	1481.28	4,12	√	√	√	√	√	√
0.4	100	1179.71	4,7,12,14	√	√	√	√	√	1182.70
	150	1355.09	4,7,12	√	√	√	√	√	1358.30
	200	1462.62	4,12	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√
0.6	100	1309.92	4,7,12	√	√	√	√	√	√
	150	1443.97	4,12	√	√	√	√	√	√
	200	1506.66	4	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√
0.8	100	1390.06	4,11	√	√	√	√	√	√
	150	1456.66	4	√	√	√	√	√	√
	200	1506.66	4	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√
1.0	100	1406.66	4	√	√	√	√	√	√
	150	1456.66	4	√	√	√	√	√	√
	200	1506.66	4	√	√	√	√	√	√
	250	1556.66	4	√	√	√	√	√	√

Table C.13: USAHLP: Comparison with other approaches, list representation, nodes = 20, CAB data

α	f	Optimal Cost	Hubs	GA Topcouglo [3] (2005)	GATS Abdnour[4] (1998)	SATLUHLP Chen. J.[11] (2007)	GA-1	GA-2	GA-3
0.2	100	***	4,7,12,14,17	967.74	967.74	967.74	967.74	967.74	977.21
	150	1174.53	4,12,17	√	√	√	√	√	√
	200	1324.53	4,12,17	√	√	√	√	√	√
	250	1474.53	4,12,17	√	√	√	√	√	√
0.4	100	1127.09	1,4,12,17	√	√	√	√	√	1142.17
	150	1297.76	4,12,17	√	√	√	√	√	√
	200	1442.56	4,17	√	√	√	√	√	√
	250	1542.56	4,17	√	√	√	√	√	√
0.6	100	1269.15	1,4,12,17	√	√	√	√	√	1270.99
	150	1406.04	4,17	√	√	√	√	√	√
	200	1506.04	4,17	√	√	√	√	√	√
	250	1570.91	6	√	√	√	√	√	√
0.8	100	1369.52	4,17	√	√	√	√	√	√
	150	1469.52	4,17	√	√	√	√	√	√
	200	1520.91	6	√	√	√	√	√	√
	250	1570.91	6	√	√	√	√	√	√
1.0	100	1410.07	4,20	√	√	√	√	√	√
	150	1470.91	6	√	√	√	√	√	√
	200	1520.91	6	√	√	√	√	√	√
	250	1570.91	6	√	√	√	√	√	√

Table C.14: USAHLP: Comparison with other approaches, list representation, nodes = 25, CAB data

α	f	Optimal Cost	Hubs	GA Topcouglo [3] (2005)	GATS Abdnour[4] (1998)	SATLUHLP Chen. J.[11] (2007)	GA-1	GA-2	GA-3
0.2	100	1029.63	4,12,17,24	√	√	√	√	√	√
	150	1217.34	4,12,17	√	√	√	√	√	√
	200	1367.34	4,12,17	√	√	√	√	√	√
	250	1500.90	12,20	√	√	√	√	√	√
0.4	100	1187.51	1,4,12,17	√	√	√	√	√	√
	150	1351.69	4,12,18	√	√	√	√	√	1353.45
	200	1501.62	12,20	√	√	√	√	√	√
	250	1601.62	12,20	√	√	√	√	√	√
0.6	100	1333.56	2,4,12	√	√	√	√	√	1333.99
	150	1483.56	2,4,12	√	√	√	√	√	√
	200	1601.20	12,20	√	√	√	√	√	√
	250	1701.20	12,20	√	√	√	√	√	√
0.8	100	1458.83	2,4,12	√	√	√	√	√	1459.74
	150	1594.08	12,20	√	√	√	√	√	√
	200	1690.57	5	√	√	√	√	√	√
	250	1740.57	5	√	√	√	√	√	√
1.0	100	1556.63	7,19	1559.19	1562.15	√	1559.19	√	1559.19
	150	1640.57	5	√	√	√	√	√	√
	200	1690.57	5	√	√	√	√	√	√
	250	1740.57	5	√	√	√	√	√	√

Appendix D

Mean of the Best Fitness Values

Following tables give the mean of the best solution values of the 30 runs i.e., average of 30 values one per run.

Table D.1: CSAHLP: AP Data, Mean of the best solution values found in the 30 runs of the GAs

		GA-1		GA-2		GA-3	
Problem	Cost	Set	List	Set	List	Set	List
10LL	224250.05	224932.34	224884.67	227673.43	226713.42	224706.43	224706.44
10LT	250992.26	253443.23	252663.23	254112.39	253221.45	254221.23	255632.45
10TL	263399.94	264322.19	263788.56	263675.56	264567.32	263971.43	265763.43
10TT	263399.94	263991.23	264201.35	263998.71	263771.21	264121.83	264536.62
20LL	234690.94	235663.23	234991.78	235221.45	234978.34	235653.45	235118.34
20LT	253517.40	254718.55	254667.72	254667.34	254321.52	254667.73	255334.42
20TL	271128.18	273441.32	272991.56	2728874.35	272887.67	273445.21	273445.66
20TT	296035.40	299313.67	298441.12	298547.09	297451.34	298434.32	298242.61
25LL	238977.95	239332.12	239667.45	239121.72	240521.32	241334.54	240332.34
25LT	276372.50	279432.14	278332.14	277551.23	277432.14	280153.52	279212.23
25TL	310317.64	315302.53	314453.23	312334.31	311653.23	315321.34	313541.23
25TT	348369.15	351375.45	352648.56	351241.29	349121.34	353421.45	352332.47
40LL	241955.71	242991.32	242554.45	244319.23	245321.33	244321.34	244765.52
40LT	272218.32	291341.76	281334.31	279431.41	274231.45	281164.54	276332.12
40TL	298919.01	302514.42	305667.31	304651.59	301339.52	302536.42	303426.86
40TT	354874.10	361311.51	357343.71	356873.33	358773.43	359342.51	360112.34
50LL	238520.59	240300.21	238919.87	2388000.34	239505.16	241334.65	244536.24
50LT	272897.49	278311.67	281700.45	27665.74	273979.57	283914.41	280655.31
50TL	319015.77	320621.41	321878.61	322451.66	323115.67	323321.54	325401.79
50TT	417440.99	423886.89	422671.56	422657.74	422287.75	422345.53	423751.15
100LL	246713.97	252800.32	249992.11	251890.39	250000.61	253412.33	253997.23
100LT	256250.41	261600.75	259141.41	258131.31	258472.19	258912.20	263439.61
100TL	362950.09	377571.62	373445.87	3687773.82	366807.31	380817.33	378667.14
100TT	474680.32	504634.53	500065.34	507644.22	5021432.43	508314.11	507422.32
200LL	241992.97	246518.43	246451.41	250887.34	252739.64	249661.32	251771.23
200LT	268894.41	277532.49	275634.12	279631.34	269621.45	276532.19	276332.31
200TL	273443.81	284305.43	291322.41	288691.52	283121.19	289341.12	28902.41
200TT	292754.97	322914.71	295755.42	294475.19	314512.67	295755.42	294141.66

Table D.2: USAHLP: AP Data, Mean of the best solution values found in 30 runs of the GAs

		GA-1		GA-2		GA-3	
Problem	Cost	Set	List	Set	List	Set	List
10L	224249.82	224721.45	224772.32	225432.61	224987.34	224991.23	224706.29
10T	263402.13	265434.23	263771.23	264551.23	264331.12	264763.43	265812.31
20L	234690.11	236771.34	236223.15	235334.31	234987.64	235443.34	235422.34
20T	271128.41	273443.45	274343.43	274334.56	274776.45	274332.67	275334.45
25L	236649.69	238443.67	238717.64	238667.43	23743.66	236797.68	237556.51
25T	295670.39	296332.12	298771.45	297331.23	296556.78	297441.34	297893.45
40L	240985.51	243112.33	241664.47	241341.45	242334.45	241660.20	242112.20
40T	293163.38	295332.45	294551.67	294332.67	294556.78	297663.67	295778.34
50L	237420.69	238667.71	238675.67	238520.58	238997.63	239518.79	238765.79
50T	300420.87	311230.45	312445.70	310143.31	305674.81	315334.41	314643.51
100L	238017.53	243199.34	242421.12	242771.34	241815.43	243492.88	244383.20
100T	305101.07	311664.23	312221.45	311332.34	312445.41	314432.45	316334.67
200L	228044.77	238651.34	236109.54	236850.09	246052.81	242635.59	241332.80
200T	233537.93	278570.91	279982.45	279848.29	279112.35	278906.80	2798678.22

Table D.3: USAHLP: Mean of the best solution values, nodes = 10, CAB data

α	f	Optimal	GA-1 Set	GA-1 List	GA-2 Set	GA-2 List	GA-3 Set	GA-3 List
0.2	100	791.93	√	√	√	√	√	√
	150	915.99	√	√	√	√	√	√
	200	1015.99	√	√	√	√	√	√
	250	1115.99	√	√	√	√	√	√
40.4	100	867.91	√	√	√	√	√	√
	150	974.30	√	√	√	√	√	√
	200	1074.30	√	√	√	√	√	√
	250	1174.30	√	√	√	√	√	√
0.6	100	932.62	√	√	√	√	√	√
	150	1032.62	√	√	√	√	√	√
	200	1131.05	√	√	√	√	√	√
	250	1181.05	√	√	√	√	√	√
0.8	100	999.94	√	√	√	√	√	√
	150	1081.05	√	√	√	√	√	√
	200	1131.05	√	√	√	√	√	√
	250	1181.05	√	√	√	√	√	√
1.0	100	1031.04	√	√	√	√	√	√
	150	1081.05	√	√	√	√	√	√
	200	1131.05	√	√	√	√	√	√
	250	1181.05	√	√	√	√	√	√

Table D.4: USAHLP: Mean of the best solution values, nodes = 15, CAB data

α	f	Optimal	GA-1 Set	GA-1 List	GA-2 Set	GA-2 List	GA-3 Set	GA-3 List
	100	1030.07	1033.45	1031.47	1030.95	1034.72	1041.31	1039.90
	150	1239.77	1241.15	1242.73	1241.19	1243.33	1247.00	1249.97
	200	1381.28	√	√	√	√	1383.21	1385.63
	250	1481.28	√	√	√	√	1491.99	1489.31
0.4	100	1179.71	1184.54	1183.27	1180.91	1182.77	1186.33	1183.29
	150	1355.09	1357.91	1356.32	1357.33	1358.77	1363.56	1361.42
	200	1462.62	√	√	√	√	1474.19	1470.34
	250	1556.66	√	√	√	√	1559.71	1562.34
0.6	100	1309.92	1312.51	1315.40	1310.11	1310.99	1311.73	1313.43
	150	1443.97	√	√	√	√	1351.61	1347.33
	200	1506.66	√	√	√	√	√	√
	250	1556.66	√	√	√	√	√	√
0.8	100	1390.06	√	√	√	√	1391.19	1393.71
	150	1456.66	√	√	√	√	√	√
	200	1506.66	√	√	√	√	√	√
	250	1556.66	√	√	√	√	√	√
1.0	100	1406.66	√	√	√	√	√	√
	150	1456.66	√	√	√	√	√	√
	200	1506.66	√	√	√	√	√	√
	250	1556.66	√	√	√	√	√	√

Table D.5: USAHLP: Mean of the best solution values, nodes = 20, CAB data

α	f	Optimal	GA-1 Set	GA-1 List	GA-2 Set	GA-2 List	GA-3 Set	GA-3 List
0.2	100	***	967.74	967.74	967.74	967.74	967.74	985.11
	150	1174.53	√	√	√	√	1178.39	1184.54
	200	1324.53	1327.11	1325.08	1325.77	1324.54	1331.10	1330.34
	250	1474.53	1478.65	1478.18	1476.01	1474.79	1478.92	1481.32
0.4	100	1127.09	1130.59	1130.59	1127.72	1131.73	1136.11	1142.10
	150	1297.76	√	√	√	√	1305.11	1311.33
	200	1442.56	√	√	√	√	1447.33	1456.66
	250	1542.56	√	√	√	√	√	√
0.6	100	1269.15	1272.83	1272.09	1270.34	1269.70	√	1273.01
	150	1406.04	√	√	√	√	√	1406.71
	200	1506.04	√	√	√	√	√	√
	250	1570.91	√	√	√	√	√	√
0.8	100	1369.52	√	√	√	√	1371.11	1373.71
	150	1469.52	√	√	√	√	√	√
	200	1520.91	√	√	√	√	√	√
	250	1570.91	√	√	√	√	√	√
1.0	100	1410.07	1415.78	1415.99	1412.43	1415.41	1417.34	1417.32
	150	1470.91	√	√	√	√	√	√
	200	1520.91	√	√	√	√	√	√
	250	1570.91	√	√	√	√	√	√

Table D.6: USAHLP: Mean of the best solution values, nodes = 25, CAB data

α	f	Optimal	GA-1 Set	GA-1 List	GA-2 Set	GA-2 List	GA-3 Set	GA-3 List
0.2	100	1029.63	1041.34	1030.04	1030.67	1031.72	1042.79	1036.71
	150	1217.34	1219.44	1217.35	1217.35	1217.35	1221.33	1227.45
	200	1367.34	1371.29	1368.15	1370.61	1368.05	1371.45	1373.44
	250	1500.90	√	√	√	√	1505.61	1509.33
0.4	100	1187.51	√	1191.73	√	1491.33	√	√
	150	1351.69	1360.66	1353.87	1353.51	1361.87	1365.11	13560.29
	200	1501.62	√	√	√	√	√	√
	250	1601.62	√	√	√	√	√	√
0.6	100	1333.56	1442.56	1334.54	1436.12	1333.66	1342.56	1333.99
	150	1483.56	1486.45	1486.04	1494.21	1486.91	1487.88	1490.03
	200	1601.20	√	√	√	√	1621.44	1610.99
	250	1701.20	√	√	√	√	√	√
0.8	100	1458.83	1478.29	1461.58	1473.41	1471.29	√	1474.08
	150	1594.08	√	√	√	√	√	√
	200	1690.57	√	√	√	√	√	√
	250	1740.57	√	√	√	√	√	√
1.0	100	1556.63	1562.67	1561.78	152.79	1561.78	√	1559.19
	150	1640.57	√	√	√	√	√	√
	200	1690.57	√	√	√	√	√	√
	250	1740.57	√	√	√	√	√	√