Using Age Layered Population Structure for the Multi-Depot Vehicle Routing Problem

Audrey Opoku-Amankwaah

Submitted in partial fulfillment
of the requirements for the degree of

Master of Science

Department of Computer Science
Brock University
St. Catharines, Ontario

Abstract

This thesis studies the NP-hard multi-depot vehicle routing problem (MDVRP) which is an extension of the classical VRP with the exception that vehicles based at one of several depots should service every customer assigned to that depot. Finding the optimal solution to MDVRP is computationally intractable for practical sized problem sets, and various meta-heuristics including genetic algorithms have been proposed in the literature. In this work, an efficient multi-population genetic algorithm based on age layered population structures for the MDVRP is proposed. Three inter-layer transfer strategies are proposed and multi-objective fitness evaluation is compared with weighted sum approach. An empirical study comparing the proposed approach with existing genetic algorithms and other meta-heuristics is carried out using well-known benchmark data. The performance found in terms of solution quality is very promising.

Acknowledgements

I will bless the Lord at all times; His praise shall continually be in my mouth. I sought the Lord and He heard me and delivered me from all my fears. (Psalm 34:1,4 KJV)

My profound gratitude is to the Lord who heard my cry and saved me from all my troubles.

I express my appreciation to my unfailing supervisors, Prof. Beatrice Ombuki-Berman and Prof. Ivo Düntsch for accepting to supervise and fund me throughout my entire stay on campus. To my other supervisory committee members, Prof Brian Ross and Prof. Michael Winter for their guidance and shared knowledge to help make this thesis a reality, I say thank you.

Special thanks to the Computer Science department and Brock University for my enrolment as well as the provision of conducive place for my study.

I thank Cale Fairchild for his responsiveness towards the provision of technical support.

For your words of encouragement, financial support and trust in me, I love you my family especially, my mum Mary Addae. I also say a very big thank you to all and sundry who have in diverse ways helped with this success story. God bless you all.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Vehicle routing problem (VRP) is a well-known optimization problem introduced in 1959 by Ramser and Dantzig [17]. This NP-hard optimization problem illustrates difficult search problems and significantly contributes to the examination of different heuristic search strategies. Though complex, VRP is significant in resolving systematic distribution procedures which help to reduce operational costs. The works of [40] and [19] present a survey on the classification and application of VRP. VRP exists in variants, and these variants have formed the basis for major discussions in available literature.

VRP is typically explained as a number of customers, each with demands to be served with a fleet of vehicles at a depot. It therefore aims at finding the minimum number of vehicles to be used to travel at a minimum distance while serving all these geographically dispersed customers. Each customer is visited once while a vehicle starts and ends at the same depot.

The variants of VRP are studied depending on given constraints. Some variants focus on just a depot with several customers basically referred to as the Single Depot VRP while others consider the time for delivery and pickup of items [12, 22, 43]. Nonetheless, this focus is less viable in recent times as most industries including soft drink industry [27] require more than one depot for their operations. This results in the application of the Multi-Depot VRP (MDVRP) - an extension of the VRP which involves a number of depots and customers. The MDVRP however, requires that any vehicle from one of many depots visits a customer exactly once. In this regard, companies within distribution and logistics management including chemical products [3] and newspaper delivery [26] have found the MDVRP useful.

This thesis focuses on the MDVRP. The MDVRP is classified into two basic types namely, the non-fixed-destination and fixed-destination. When using the fixed-destination MDVRP approach, a vehicle starts and ends a route at the same depot but when applying the non-fixed-destination MDVRP approach, a vehicle starts and ends a route at different depots.

The MDVRP is computationally intractable and various approaches have been proposed in the literature to help solve this problem. Exact methods as recently reported by Contardo [14] are scarce; even though useful, this approach consumes relatively much computation time even when applied to problems with smaller sizes. However, a number of heuristics including Tabu search [13, 47] as well as adaptive large neighborhood search (ALNS) [46] have tackled the MDVRP with success. Nonetheless, in order to find a fast and near optimal solutions to the MDVRP, a number of meta-heuristics algorithms have been implemented. Notable among them is the genetic algorithm (GA).

GA is a population based meta-heuristics that follows the Darwinian processes of natural evolution including selection of fitter individuals, recombination and mutation. An analysis of existing research on the MDVRP with GA shows that little research focused on minimizing both number of vehicles and total distance which is the multi-objective optimization (MOP) approach, while the rest focused on minimizing only the total distance.

A previous study given in [38] reveals that GAs performance are less optimal compared to other meta-heuristics. That notwithstanding, these MOPs provide more room for decision making based on preference while compromises are made. It has also been established that "Minimizing the number of vehicles affects vehicle and labour costs, while minimizing distance affects time and fuel resources" [44]. It is therefore plausible to have an algorithm that helps in minimizing both objectives simultaneously. GA is prone to settling on suboptimal solutions hence resulting in the problem of premature convergence.

Age Layered Population Structure (ALPS) [32] is a kind of multi-population evolutionary algorithm that resolves the problem of premature convergence in algorithms which display characteristics of randomness. Some advantages associated with using ALPS are found in[1, 32, 33, 34, 48]. Through a routine initiation of new individuals into the population, ALPS increases the likelihood of an evolutionary algorithm which is hardly converged on a local optimum but consistently searches various portions of the fitness landscape with greater opportunity to find a global optimum solution. Using ALPS based GA, this thesis focuses on the fixed-destination MDVRP with both route and capacity constraints. In this variant of MDVRP, the number and locations of both depots and customers are known a priori.

Our ALPS-GA employs a simple permutation of randomly clustered individuals through the use of a route scheduler [44] with a simple mutation to reassign individuals from one depot to the other. The proposed ALPS-GA is compared to Thangiah *et al.*'s [52] genetic clustering as well as Ombuki *et al.*'s [44] genetic algorithms for the MDVRP. The proposed ALPS-GA performed better than that of Ombuki *et al.* [44] and Thangiah *et al.* [52]. Further MOP, a comparison with normalized sum of ranks was done. This fitness evaluation strategy equally proved efficient with the ALPS-GA.

1.1 Summary of the Main Contributions

This thesis made the following main contributions;

1. Proposed an ALPS-GA for the fixed-destination MDVRP using multi-objective approach that minimized both the number of vehicles and total distance travelled.
2. Proposed three inter-layer transfer strategies and determined their effectiveness.
3. Proposed a reassigning mutation.
4. Performed an empirical study comparing the proposed ALPS-GA with existing meta-heuristics.

1.2 Structure of Thesis

Chapter 2 gives a background study on the fundamental components of the thesis including ALPS, GA, and multi-objective optimization with its accompanying fitness evaluation methods and presents a literature review on related works. Chapter 3 provides the details of the proposed ALPS based GA. Chapter 4 presents experimental results. Discussions with concluding remarks and future works given in Chapter 5.

Chapter 2

Background

This chapter provides background information on MDVRP, ALPS and its applications, and genetic algorithms and multi-objective optimisation.

2.1 Multi-Depot Vehicle Routing Problem

Within a typical distribution system, assume that the customer size, location, individual customer demands as well as the number and location of all potential depots are known while vehicle type and size are also given *a priori*. We directly employ the MDVRP model adopted by Renaud *et al* [47]. Let $G = (V, A)$ be a directed graph, where V is the vertex set, and A is the arc set. The vertex set V is further divided into two disjoint subsets $V = V_{cus} \cup V_{dep}$ where $V_{cus} = \{v_0, v_1, v_2, \ldots, v_n\}$ represents the set of customers, and $V_{dep} = \{v_{n+1}, \ldots, v_{n+d}\}$, the set of depots. We align each customer to a non-negative demand d_i and a service time ρ_i for $v_i \in V_{cus}$. We use the arc set A to denote all available connections between nodes including those denoting depots. We define a cost matrix $C = (c_{mk})$ on A to represent travel times. We use travel times to mean Euclidean distance as termed in other publications on MDVRP. We emphasize on problems for which C is symmetric and solves the triangle inequality, i.e., $c_{kl} = c_{lk}$ and $c_{mk} \leqslant c_{ml} + c_{lk}$, for all m, l, k , where c_{kl} is the distance from customer k to customer l.

For each depot, $v_k \in V_{dep}$ with $n + 1 \leq k \leq n + d$, t_k represents the number of identical vehicles with capacity Q at v_k. This is depicted in Figure 2.1 with two depots and nine customers.

Objectives:

1. Minimize the total number of vehicles used to serve all customers.

2. Minimize the total distance travelled by all the vehicles.

Constraints to be observed:

- A fleet of same size vehicles with equal capacities Q is located at each depot.

- Each vehicle begins and terminates at the same depot. This is termed as fixed destination problem.

- All the customers and depots with their capacities, demands and locations are known in advance.

- The route duration spent by a vehicle does not exceed a preset limit.

- Customers of a given route have a total demand less than or equal to the total capacity of the vehicles assigned to that route.

- No customer has a demand value more than the capacity of a vehicle.

- Lastly, each customer is visited once and only once by a vehicle.

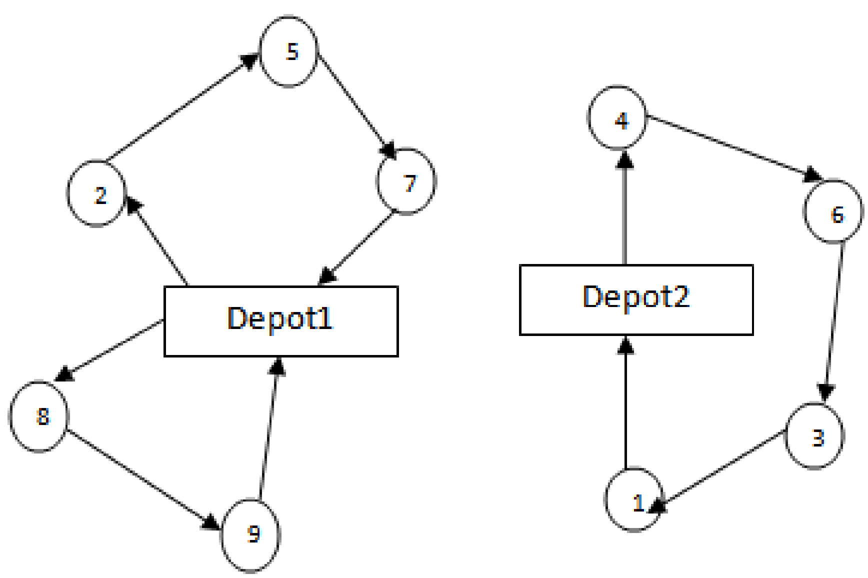

Figure 2.1: An example of MDVRP with 9 customers and 2 depots.

2.1.1 Previous Work on MDVRP

Cardon *et al.* [20] introduced a polynomial time approximation to solve both fixed and non-fixed destination (starts a route from a depot and end at a different depot) variants of the MDVRP. The run time of the solution was improved by eliminating the double exponential term in the run time which was assumed to be caused by computing the optimal tour. Contardo and Rafael [14] proposed vehicle flow and set partitioning formulations to MDVRP. The algorithm used both cutting planes method and the column and cut generation. Baldacci and Mingozzi [2] also proposed an exact method based on LP relaxation and Lagrangian relaxation to reduce the number of variables in the formulation and then applied integer linear programming to the problem. Laporte *et al.* [39] also proposed a branch and bound algorithm but tested for just some small instances of the MDVRP.

Some heuristics like savings criterion which incorporates multiphase heuristics was used by Chao *et al.* [13] to solve the MDVRP to detect new best solutions. Renaud *et al.* [47] introduced FIND as an improved tabu search with a post-optimization strategy to the solution. FIND performed the fast improvement (FI) where it considered inter, intra and 3-routes exchange. Two tabu search algorithms were proposed [15, 47] where [15] was designed such that it used fewer user controlled parameters. Cordeau *et al.* proposed another algorithm using both tabu search and Integer Programming [16].

Thangiah *et al.* [52] proposed genetic algorithm with adaptive clustering methods and post-optimization strategies for the MDVRP. Nilay *et al.* [56], proposed an algorithm which served as an improvement on Thangiah *et al.*'s work by considering other clustering methods. Ombuki *et al.* [44], proposed a genetic algorithm which used a simple fast approach to assign customers to their nearest depots and applied an efficient crossover approach with local search like properties. Surekha *et al.* [51] proposed another genetic algorithm which employed the grouping of customer to their nearest depots and a routing strategy that used Clarke and Wright saving method [23]. A survey in [38] provides a summary of genetic algorithm strategies used in solving MDVRP.

Liu and Yu (2013) proposed a hybrid algorithm using ant colony (ACO) and GA. The GA optimized the parameters for the ACO [41]. Ropke *et al.* [46] proposed an adaptive large neighborhood search (ALNS) which provided some of the best MDVRP results.

A population based evolutionary search combined with aggressive improvement capabilities of neighborhood metaheuristics approach was proposed [54]. Gilbert and

Johnson's sweep algorithm [24] was followed in [50] to solve MDVRP.

There have been other hybrid algorithms proposed to solve the MDVRP. Jeon *et al.* [36] suggested hybrid genetic algorithm (HGA) for MDVRP, which had an improvement of generation for the initial solution and a float mutation rate for escaping from the local solution. Ho *et al.* [2008] also proposed HGA1 and HGA2 to solve MDVRP [29]. A comprehensive review on MDVRP can be seen in [44]. Parallel Improved ACO (PIACO) [55] considered MDVRP as a single depot VRP by introducing a virtual depot which linked all the available depots from which a route starts and terminates. Well noted from this paper is also the fact that it considered subpopulating the individuals (pheronomes). They used the coarse-grained strategy to perform inter subpopulation exchange of the pheronomes using the ring topology to prevent premature convergence. One recent paper on MDVRP [11] introduced the use of multi-population strategy for MDVRP and introduced diversity based on exchange of individuals based on fitness.

2.2 Age Layered Population Structure

Hornby [32] proposed the Age-Layered Population Structure (ALPS) as a technique for handling the problem of premature convergence in evolutionary algorithms (EAs). Hornby [32] reviewed some multi-population algorithms such as the hierarchical fair competition (HFC) and adaptive hierarchical fair competition (AHFC). He addressed the weaknesses of these algorithms by proposing the ALPS to be an algorithm with an age attribute for an individual. The downside of the two aforementioned algorithms resulted from the transfer of individuals from one layer to the other based on fitness. The superiority of individuals with better fitness are highly observed because individuals with low fitness are mostly not allowed to be transferred. However, ALPS measures the ageing of genetic material in the evolution of a population.

In ALPS, age is used as a factor to restrict competition and breeding in the population of individuals. In using age as a factor to restrict breeding, it decreases the likelihood of very fit old individuals controlling the evolution process, which mostly leads to early convergence in traditional EA. ALPS algorithm implementation is different from traditional EAs as it requires new parameters, such as how many solutions to keep in each layer and how to pick cutoffs for the age layers, etc. ALPS groups individuals into age-layers and introduces new randomly generated individuals on a regular basis into the youngest (layer 0) layer. The individuals at each layer are allowed to develop and evolve in parallel while inter layer transfer of individuals take place. A feature of ALPS is that, there is no age limit for the last layer to retain all the best individuals. This retention of best individuals though depends on the type of inter-layer transfer used. The outcome of introducing randomly generated individuals is, an evolutionary algorithm (EA) that continuously searches new parts of the fitness landscape. The ALPS approach in EA helps to overcome the problem of early convergence in the algorithm as evident in [32, 33, 34]. The search behavior of ALPS shows that, randomly generated offspring are able to change the population out of mediocre local optima in the fit portions of the fitness landscape. This benefit is also achieved through the breeding that happens among the layers. A clustered population around a different fitness landscape is created with each restart of the bottom layer in ALPS; hence the resulting ALPS population increases the exploration of the fitness landscape. An individual moves into the next higher layer when it attains its accepted maximum age in the layer [1]. An individual in the next higher layer is discarded to give chance to a new individual for a constant population to be maintained in all layers.

2.2.1 ALPS Algorithm

The algorithm starts with randomly initializing the first layer. Through the evolution process, the other layers are filled. The bottom layer is replaced by newly generated individuals and the old generation moved to the next layers or discarded [32].

Algorithm 1 Pseudocode for ALPS

```
procedure ALPS()
    Read parameter file
    Get number of layers, age gap, AgeingScheme
    setLayers ⟸ InitialiseLayers(number of layers, age gap, ageingScheme)
    while !termination criterion do
        if (generation == 0||generation%layer0.agelimit == 0) then
            M ⟸ RandomlyCreateNewIndividuals()
            Else
            evolveIndividuals()
        end if
        if (generation>layer0.agelimit) then
            evolveIndividuals()
        end if
    end while
end procedure
```

```
procedure evolveIndividuals()
    Perform elitism
    procedure PerformSelection()
        if (layer0) then
            Parents ⟸ selectparents(layer0)
            Else
            Parents ⟸ selectParents(currentLayer(C), (C-1)Layer, selection pressure)
        end if
    end procedure
    Perform Crossover
    Perform mutation
    if (individual.age==layer.ageLimit) then
        nextLayer ⟸ moveIndividual()
    end if
end procedure
```

2.2.2 How to measure Age

Age has been used as a factor in various EA systems to improve performance [31, 35]. The starting age for all individuals is set to 1 in all these systems. This is independent of how they were created, that is whether through crossover, mutation or through random creation. The difference between these systems and ALPS is, ALPS provides age to an individual based on how long its ancestors have existed while the former increases the age of an unchanged individual in terms of genes. Age is the degree of how long an individuals genotypic material has existed in the population. This is to say that an individual in ALPS attains its age from the age of the highest parent plus one [32]. There are a number of ageing schemes that are used in ALPS namely linear, polynomial, fibonacci, or exponential [32] as shown in Table 2.1. Based on user decision or empirical tests, a user selects an ageing scheme for the experiments. In setting up an ALPS run, the user sets out parameters such as the age gap for each layer and the number of age layers. The selected ageing scheme with increasing limits is multiplied by an age-gap parameter per layer, which helps in controlling the population size, the age limit per layer and the number of layers since there is normally a need to push up individuals from one layer to the other.

2.2.3 Related Works to the ALPS

Hornby in [33] revisited the idea of measuring the age of an individual and determined that by using age to restrict competition and breeding, the population is not over-

Table 2.1: Ageing Schemes for ALPS [32]

Aging-Scheme	0	1	2	3	4
Linear	1	2	3	4	5
Fibonacci	1	2	3	5	8
Polynomial(n^2)	1	2	4	9	16
Exponential(2^n)	1	2	4	8	16

taken by highly fit individuals because younger individuals are allowed to compete with older ones and are also free to develop. Age was explained to be measured by how long an individual's ancestors have existed contrary to other age-based algorithms. These other age-based algorithms rather measured age by how long an individual has maintained the same genotypic material. Hornby showed the efficiency of ALPS with GA on the Black-Box Optimization Benchmarking (BBOB) problems and obtained better results. The ALPS had been used in the genetic programming (GP) context but Hornby [33] considered it within GA and proposed that ALPS can be used for any EA. In [34], Hornby, compared a steady state GA based on ALPS with differential evolution (DE) and a covariance matrix adaptation evolution strategy (CMA-ES). Hornby *et al.* [7] used ALPS and early stopping strategies to further increase diversity while decreasing search time.

In [49], ALPS enhanced cartesian genetic programming (CGP) was declared better than the traditional CGP when compared on an image operator problem.

Additionally, the applicability and efficiency of ALPS exhibited as it performed positively on a financial portfolio problem studied [45].

A comparative study on canonical GP, ALPS and feature selection ALPS (FSALPS) [1] on a feature selection problem has been conducted.

Spatial coevolution (SCALP) [28] focused on reducing the increase of the tree size without any improvement in fitness usually referred to as bloat.

2.3 Genetic Algorithm

John Holland introduced the genetic algorithm (GA) as a meta-heuristics which models a given problem by following the natural processes of evolution [30]. In a GA, a population of individuals (i.e. chromosomes) representing potential solutions for the problem at hand is transformed into a new population using the Darwinian principle of survival of the fittest and natural selection. A new population of individuals is constantly generated with the aim of getting the best solution to the problem until reaching a termination criterion. GAs do not guarantee finding an optimal solution to a given problem but often obtain good or near optimal solutions to many problems in feasible time [38]. GAs have been considered for problems like the hub location problems [42], multi-objective management [37] and several vehicle routing problems [12, 43, 44, 50, 51].

Algorithm 2 shows an outline of a simple GA.

Algorithm 2 Pseudocode for GA

```
procedure SimpleGA()
    Input dataset, parameters
    Randomly generate initial population
    while !(termination criterion) do
        Compute chromosome fitness
        Select elite population
        Select 2 individuals as parents
        offsprings ⟸ perform crossover(parents)
        perform mutation(offsprings)
        newPopulation ⟸ individuals
        Population ⟸ newpopulation
    end while
```

2.3.1 Chromosome Representation and Initial Population Generation

A chromosome represents a probable solution to a problem at hand. GA models the chromosome in a form of an efficiently designed data structure. Floating point numbers, integers, binary strings, order-based representations, set-based representations, among others are examples of representations existing in literature [42], [9]. A number of these chromosomes are randomly created to form the initial population.

2.3.2 Fitness Evaluation

The fitness value of each individual in the population is calculated using a function evaluation that measure quality of a solution for a problem at hand. The values produced by the fitness function show the cost of the solutions of the population in a generation and deliver a source for the identification of solutions that are fitter for the succeeding selection process.

2.3.3 Recombination

Through crossover, two selected individuals from the population are paired up for reproduction to yield offspring by exchanging their genes.

2.3.4 Mutation

Mutation potentially introduces a genetic diversity from one generation of a population to the next and helps the algorithm to avoid getting trapped in a local optimum. The genes of the chromosomes are mutated after crossover to alter their composition. In mutation, only one chromosome is selected and the resulting offspring is evaluated again and added to the new population to continue the evolution process.

2.3.5 Elitism

Elitism involves introducing the best individuals in the current generation into the next generation unchanged. Though the elite population can be denied of selection, these individuals are allowed to be selected as parents for recombination and this gives an assurance that the quality of the solution will not drop from one generation to the other.

2.3.6 Selection

Selection is a stage of a GA in which an individual is chosen from a population for a later breeding using recombination, mutation. Examples of selection mechanisms include tournament selection , roulette wheel, scaling selection, and rank selection.

2.3.7 Termination

Below are some termination criteria in a GA algorithm.

- A termination criterion can be when the algorithm reaches a pre-fixed number of generations.
- It can also be when a chromosome reaches a particular fitness level. .

2.4 Multi-Objective Optimization (MOP)

MOP is a problem instance which derives the optimal solution from two or more objectives. Oftentimes some compromises are made on these objectives to reach a decision. Success in the use of MOP have been shown in [8, 18, 21]. An example of such a problem could simply be realized in the field of transportation. While the company aims at securing a big vehicle, it also considers fuel consumption to be important. The company might end up buying something not as big as it considered based on the fuel consumption. As this thesis aims at minimizing the number of vehicles as well as the total distance travelled, the problem is viewed as a multi-objective optimization hence in determining the fitness of an individual; some known fitness evaluation strategies are implemented.

2.4.1 Weighted Sum

This fitness evaluation technique introduces a bias by requiring that some objectives be given priority. The disadvantage of this approach is that determining the appropriate weights for the objectives becomes difficult and time consuming. Furthermore this approach converts a multi-objective problem to a single objective one. For example assuming there is an individual X with x_1, x_2, $\cdots$, x_n as the n objectives to be evaluated and $w_1, w_2, \cdots, w_n$ as the weights corresponding to each objective, the fitness value for individual X is given by

$$Fitness(X) = x_1 * w_1 + x_2 * w_2 + \cdots + x_n * w_n \tag{2.1}$$

2.4.2 Pareto Ranking

Pareto ranking employs the use of dominance to give ranks to individuals which will serve as fitness [53]. It gives better ranks priority to individuals whose fitness values per objectives can not be categorically outperformed by others.

Basically the whole idea of pareto ranking is to maintain the independence of each of the objectives considered for a candidate solution as opposed to weighted sum which merges the objectives together. Having maintained the individuality of the objectives, they are used collectively to stratify the population into groups based on dominance. This is to say that, the candidates in a stratified group can be said to have equal advantage or strength as none of them is better than the other. Hence the members in a stratified group are only better than members in another group beneath them. This idea is important because, it can be difficult to compare some objectives. For example consider the speed of a car verses the car's safety. This evaluation provides room for decision makers to make selection based on what they actually are interested in. We now explain how pareto ranking is performed given a number of individuals in a population for our minimization problem.

For a problem defined by a vector of objectives $\vec{f} = (f_1, ..., f_n)$ which is subject to appropriate problem constraints, we say $\vec{u}$ dominates $\vec{v}$, $(\vec{u} \preceq \vec{v})$ iff $\forall i \in (1, ..., n) : u_i \leq v_i \wedge \exists i \in (1, ..., n) : u_i < v_i$. This by definition means, a vector is dominated if and only if another existing vector is better in at least one objective and at least as good in the remaining objectives.

For the candidates in a population, compute the fitnesses of individuals and convert to ranks. Select all the non dominated individuals and assign rank 1 to them as their fitness. These individuals are eliminated from the population and similarly the rank 2 individuals are selected and eliminated. Subsequent selection,ranking and elimination are performed until all the individuals are ranked. Each time individuals are selected, they all have the same rank number and this is calculated as currentRank=previousRank +1. These ranks are now used as the fitnesses for the individuals to undertake the other processes such as elitism, tournament selection and evolution in general. The pseudocode for pareto ranking is given in Algorithm 3 below.

Algorithm 3 Pseudocode for Pareto Ranking [44]

```
procedure PARETORANK()
    Input Population
    procedure PERFORMPARETORANKING()
        currentRank ⟸ 1
        Popsize ⟸ populationSize
        N ⟸ Popsize
        while (N ⩾ 1) do
          for (i=1 to N) do
            if v⃗_i is non-dominated then
                rank(v⃗_i)= currentRank
            end if
          end for
          for (j=1 to N) do
            if rank(v⃗_j)= currentRank then
                remove (v⃗_j) from population
                Popsize=Popsize-1
            end if
          end for
          currentRank=currentRank+1
          N=Popsize
      end while
  end procedure
```

2.4.3 Sum of Ranks

Bentley and Wakefield [5] proposed this fitness evaluation method basically to solve the problem of outliers in Pareto ranking. This method has been noted for problems with many objectives but has also been proven by some researchers to work well for problems with few objectives. Notable is the work by Bergen and Ross [6] which considers three (3) attributes of a color gradient distribution. There are two types of sum of ranks in existence. Both of them are explained below with Figure 2.2 showing the one implemented in this thesis.

Given a set of individuals each with multiple objectives, first compute the fitness per objective for each individual. Rank each objective per individual in the population, sum up the ranks for all the objectives per individual and perform re-ranking

The aforementioned steps outline the first type of sum of ranks. The normalized sum of ranks was important because often it happens that the total ranks of per objective are less than the total for other objectives which could lead to unfair distribution of overall ranks to the individuals in the population. Hence, immediately after the ranks for each objective is computed, one finds get the highest rank number from the ranks of each objective. Then we divide each rank of each individual per objective by the highest rank number per objective (n), sum the result from n for each individual and re-rank the individuals and set the rank as the fitness. This balances contribution of each objective to the overall score.

	Objective 1			Objective 2			Sum of
Individual	Raw Score	Rank	Normalised	Raw Score	Rank	Normalised	Normalised Ranks
1	300	2	1	50	2	0.4	1.4
2	60	1	0.5	200	5	1	1.5
3	300	2	1	30	1	0.2	1.1
4	300	2	1	80	3	0.6	1.6
5	300	2	1	100	4	0.8	1.6

Figure 2.2: Normalised sum of ranks for a minimization problem with two objectives.

Chapter 3

ALPS based GA for the MDVRP

The implementation details of the proposed ALPS-GA for the MDVRP is provided here. Generational ALPS was utilized with GA serving as the evolutionary algorithm. An overview of the ALPS algorithm is illustrated in Chapter 2 (Algorithm 1) and an outline of the proposed system is given in Figure 3.1.

ALPS-GA starts by designing the layers and setting the age limits for each layer. The GA is activated at the lower layer which performs a simple clustering of customers to their nearest depots. The assigned customers are effected upon by the route scheduler to produce viable chromosomes randomly for initial population into the first layer. After subsequent evolutions involving the use of tournament selection which retain elite population [25], the use of best cost route crossover (BCRC) [44] and re-assigning mutation, these chromosomes reach their age limit and are transferred to the next layer with the reintroduction of new individuals into the lower layer. The process is repeated until all layers and termination criterion are reached.

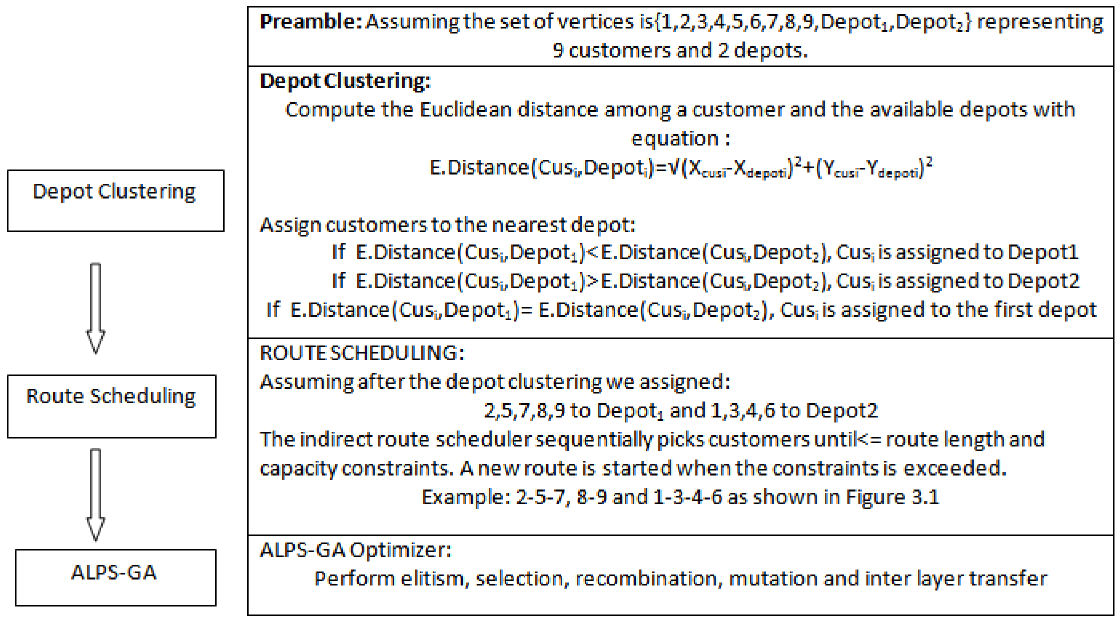

Figure 3.1: Processes involved in finding optimum solution.

3.1 Initial Depot-Clustering

The Euclidean distance is used to assign customers to their nearest respective depots. In [44] some of customers were determined based on how close they were to more than one depot. Similarly to the work in [44] these borderline customers are used for mutation in this thesis.

3.2 Chromosome Representation and Initial Population

An indirect representation of chromosomes implemented depicts the fleet of vehicles used and the order used to traverse the customers. A path representation which involves the use of an intelligent route scheduler is employed to produce a n-vector cluster of routes with n being the number of depots. Figure 3.2 depicts the chromosome representation. This figure illustrates an MDVRP with 10 customers and two depots,where 6 and 4 customers are assigned to the individual depots respectively. The initial population consists of only feasible candidates.

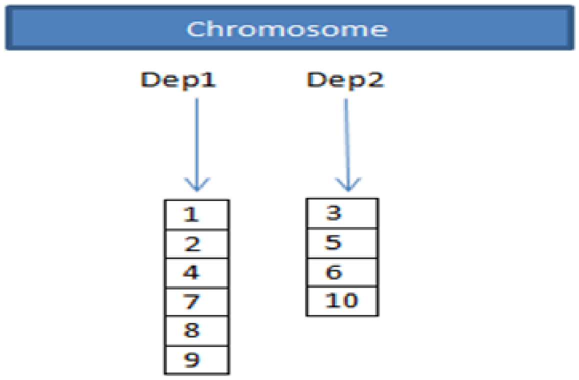

Figure 3.2: Chromosome Representation with two depots and 10 customers with no delimiter showing start and end of route per depot.

3.3 Route Scheduler

Since an indirect chromosome representation is used, the purpose of the route scheduler is to convert a random permutation of customers per depot into feasible routes.

3.4 Fitness Evaluation

3.4.1 Weighted Sum

Weighted sum involves associating weights to the individual objectives and computing the total fitness as a single objective problem. For the MDVRP, an aim was to minimize the total number of vehicles and total distance. An individual ind is mathematically represented as $ind = (A, h : A \rightarrow V_{dep}, \gamma : A \rightarrow V_{cus}{}^{*})$, where A is the set of vehicles, h assigns a home depot to each vehicle, and γ assigns a route, i.e. a sequence of customers to each vehicle. The distance dist(d,r) of a route $r = cus_1, \ldots, cus_n$ starting at depot d can be computed by $dist\{d, cus_i, \ldots, cus_n\} = c_{d,cus_1} + c_{cus_1,cus_2} +, \ldots, +c_{cus_{n-1},cus_n} + c_{cus_n,d}$. The fitness of the individual is found by using the equation:

$$Fitness(ind) = \alpha * |A| + \beta * \sum_{a \varepsilon A} dist(h(a), \gamma(a)) \quad (3.1)$$

α and β are the weights associated with the number of vehicles and the total distance respectively with $dist(h(a), \gamma(a))$ representing total distance travelled at a depot. Since this problem had an inverse relation, we empirically established α as 100 and β as 0.0001.

3.4.2 Pareto Ranking

First the total vehicles and distance of each individual were computed. Both scores were collectively used to rank the individual. These ranks therefore served as the fitness of the individual following the algorithm in Section 2.4.2.

3.4.3 Sum of Ranks

This thesis followed the explanation of normalised sum of ranks given in Section 2.2. Both objectives were individually evaluated first and also underwent the normalization process. The ranks were assigned as the fitness of the individuals.

3.5 Elitism

The ALPS-GA implement elitism the same way as the traditional GA. It selects all elite population from the current layer. These selected individuals are retained in the old population and could be reselected for recombination and mutation. Due to the inter layer transfer process, all but the last layer retain these elite individuals until the layer's age limit is reached. The elite population size is set to 3 as shown in Table 4.3.

3.6 Selection

The k size tournament selection was used. K was set to 4 of the population per layer. The individuals are selected from the current layer and the layer one level below. There is a selection pressure empirically established such that 80% of the individuals are selected from the current layer while the remaining 20% are selected from the lower layer. This procedure does not affect the first layer (Layer 0) and thus all the 4 individuals are selected from layer 0. We used a condition that was aimed at reducing the search of an individual from the lower layer. We sorted the population based on fitness and searched through the first quarter of the sorted population. After getting the individuals for the tournament competition, we picked the one with the least fitness as the winner.

3.7 Crossover

The Best-Cost Route Crossover (BCRC) [44] is employed in this thesis. It was introduced in [43] as a problem specific crossover to solve the VRP with Time window (VRPTW). BCRC has been used for variants of VRP [10, 44] and Hub Location problem [42]. In [44], the algorithm was slightly changed to solve the MDVRP problem. Figure 3.3 depicts the implementation of BCRC and the following steps are taken to perform BCRC.

- Randomly select two parents from a given set of viably created population as P1 and P2.
- Randomly select a route from each parent at a selected depot. For example route with customers 2,4 from P1 and customers 8,5 from P2.
- Remove from P2 the customers in the route from P1 (2,4) and from P2 the customers 8,5.
- Re-insert the customers removed from each parent to form the offspring.
  - Pick a removed customer from a parent (eg. customer 8 from P1) and insert at each index in an unremoved route in the parent at the selected depot.
  - Compute the cost of insertion at the index, track its feasibility status, and create an ascending ordered list of (index, feasibility and cost) based on the cost. If it exceeds the constraint and breaks a route into two, infeasibility is true.
  - At a given probability (0.8 used in this thesis), randomly generate a number r between 0 and 1.
  - If r $\leq$ given probability, insert at the first feasible index, or create new route with only the customer if no feasible index exist.
  - If r $>$ given probability, select the index at the first position in the order list for insertion irrespective of the feasibility status.

3.8 Mutation

The re-assigning mutation used in this thesis was inspired by the single customer rerouting and the adaptive inter depot mutation in [44].

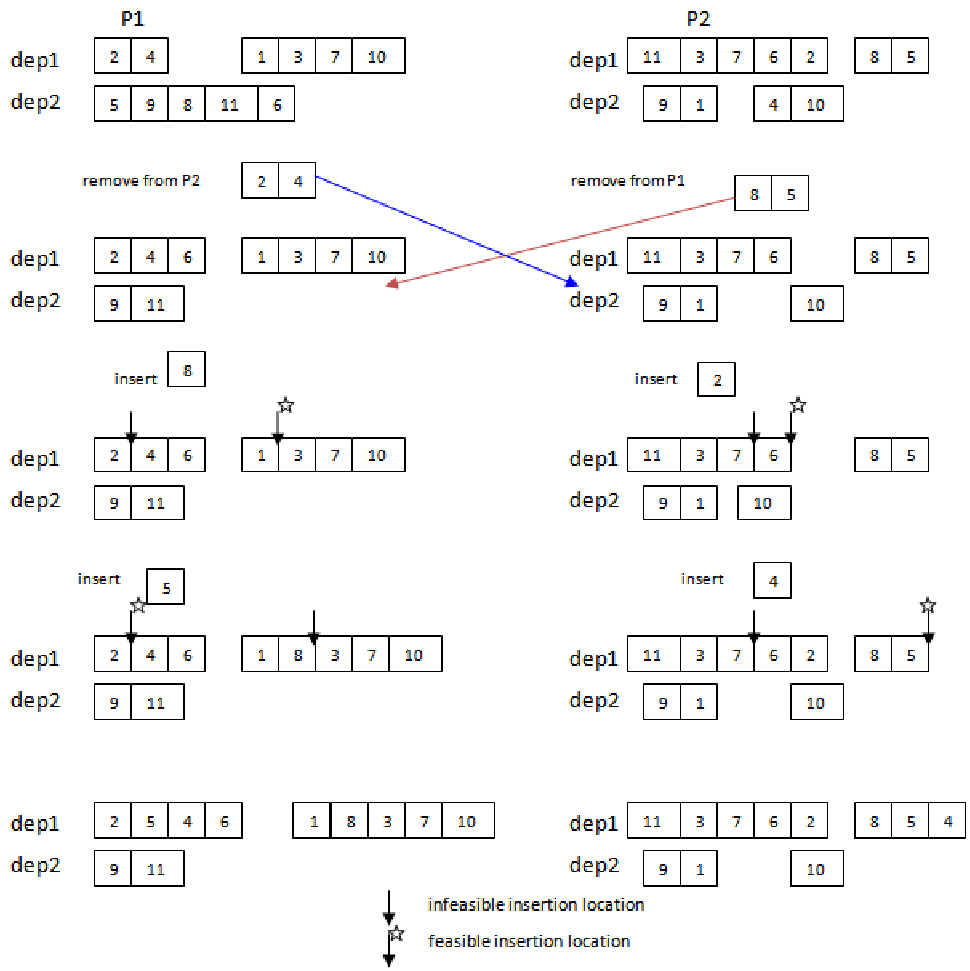

Figure 3.3: Best Cost Route Crossover.

Single customer rerouting: Select a customer randomly from a depot and reinsert the customer at a random position within the same depot. For example: Given 2,7,8,9,10 as customers at depot 1 of a chromosome, if customer 8 is selected, it could be reinserted with the resulting order of customers at depot1 as 2,7,9,10,8

Inter depot mutation: During initial clustering of customers to the depots, a list of pairs of customer and depots referred to as candidate swappable customer list is built where the following inequality holds:

$$\frac{(c_{cu,d}) - min}{min} \leq bound \tag{3.2}$$

The euclidean distance from a customer to a depot is given by $c_{cu,d}$ and *min* is the distance from a customer to its closest depot and *bound* is a value empirically determined as 3. These customers are termed as borderline customers and are regarded as having the ability to reassign to other depots than the currently assigned depot during the evolutionary process.

Re-assigning Mutation: Create the swappable customer list from a customer with its allowed depots (the initial assigned depot inclusive).

- Select a customer and a depot from the swappable customer list for reassignment.
    - We achieve single customer rerouting if the depot selected is the currently assigned depot.
    - It is inter depot if it selects a different depot from the current one.

3.9 Replacement Strategy

The ALPS-GA used the generational replacement strategy for evolution which replaces the entire current population with a new one containing the offspring and elite population.

3.10 Inter Layer Migration/ Transfer

During the evolutionary process, the allowable age of the individuals at the layers except the last layer must be checked. Whenever the individuals reach their age limit layer, these individuals are to be migrated to the layer immediately above this current layer. How these individuals are being transferred/ migrated affect the result of the algorithm. Three replacement techniques were implemented are as follows:

3.10.1 Reverse Tournament Worst (RTW)

Individuals to the tournament size given are selected from the immediate higher layer ($layer_h$) to the current layer ($layer_c$) for competition [1]. The worst individual in terms of fitness is selected and replaced. This replacement technique keeps the best individual in ($layer_h$).

3.10.2 Best Tournament Replacement (BTR)

We selected some individuals and performed tournament selection. The best individual was replaced by the aged individual from the lower layer. The problem with this approach was, because the very best individual from ($layer_h$) was not exempted from the tournament selection, this individual could be replaced. Replacing this best individual dragged the search back to either starting or stuck on a different result which could be worse.

3.10.3 Best Individual Replacement(BIR)

The aged individuals are added to the individuals in ($layer_h$). The combined individuals are sorted based on fitness. Individuals to the layer's population size are selected in ascending order from the sorted individuals. The unselected individuals are eliminated from the ($layer_h$). This technique ensures that the best individual in ($layer_h$) is always kept.

Chapter 4

Experimental Results

This chapter provides the experimental details and results of the performance of the proposed ALPS-GA on the well known MDVRP datasets [57].

4.1 MDVRP Dataset

The MDVRP dataset is made up of 23 instances from P01-P23. P01-P07 have 50 to 100 customers and 2 to 5 depots. P08-P11 have 249 customers and 2 to 5 depots. The last twelve have customers between 60 and 360. In all instances, a fleet of vehicles were given with homogeneous capacity per instance. The capacities range between 60 and 500. The 23 dataset is categorized into two. One category has only capacity constraint and the other with both capacity and route length constraints. The route length constraint is a preset total distance a vehicle should not exceed when forming routes. The customers and depots all lie on a Euclidean plane hence the preset route length is given as the total Euclidean distance a vehicle can travel. All the MDVRP dataset instances are shown in Table 4.1. We downloaded these instances from [57].

Table 4.1: Dataset for the MDVRP Experiment

Instances	Customers	Capacity	Depots	Instances	Customers	Capacity	Depots
P01	50	80	4	P13	80	60	2
P02	50	160	4	P14	80	60	2
P03	75	140	5	P15	160	60	4
P04	100	100	2	P16	160	60	4
P05	100	200	2	P17	160	60	4
P06	100	100	3	P18	240	60	6
P07	100	100	4	P19	240	60	6
P08	249	500	2	P20	240	60	6
P09	249	500	3	P21	360	60	9
P10	249	500	4	P22	360	60	9
P11	249	500	5	P23	360	60	9
P12	80	60	2				

4.2 Experimental Setup

The proposed ALPS-GA was implemented in Java 1.7 with 30 runs per experiment performed on Intel(R) Core(TM) i5.3570 cpu @ 3.30GHz with 8GB RAM on Ubuntu 15.04 environment. A number of experiments were performed as follows.

1. Accuracy comparison between our non-ALPS based GA and known GAs.
2. A comparative study between the proposed ALPS-GA and our non-ALPS based GA using weighted sum.
3. A comparative study on the three inter-layer transfer strategies implemented using the weighted sum.
4. A comparative study among the weighted sum and the two multi-objective fitness evaluation strategies.
5. A comparative study among ALPS-GA and known GAs.
6. Accuracy comparison among ALPS-GA and known non-GA approaches using weighted sum.

4.2.1 Parameters

The empirically established parameters are given in Table 4.2 and Table 4.3.

Table 4.2: Parameter setting for non-ALPS based GA Experiment

Parameters	Value
Number of runs	30
Replacement	Generational
Population size	400
Generation span	3000
Selection	tournament $k = 4$
Crossover rate	0.8
Mutation rate	0.2
Elite size	3

Table 4.3: Parameter Setting for the ALPS-GA Experiment

Parameters	Value
Number of runs	30
Replacement	Generational
Population size per layer	80
Generation span	3000
Selection	tournament $k = 4$
Crossover rate	0.8
Mutation rate	0.2
Elite size	3
Number of Layers	5
Ageing Scheme	Polynomial
Age Gap	50

We show how we empirically determined the given parameter settings for the ALPS-GA in terms of ageing scheme and Age gap in the Appendix A.

4.2.2 Comparison of non-ALPS based GA with known GAs

Tables 4.4 and 4.5 show the comparative results with Thangiah *et al.* [52] and Ombuki *et al.* [44]. Comparisons are only done if the total number of vehicles used are the same. The boldfaced values indicate the best result in 30 runs. From Table 4.4, it is seen that both non-ALPS based GA and Thangiah *et al.*'s performance is comparable.

From Table 4.5, it is seen that our GA reduced the number of vehicles by 1 in two instances. It also reduced one more than Ombuki *et al.* [44] in terms of distance.

Table 4.6 showed overall performance which looked at instances with the same number of vehicles. It is shown that Thangiah *et al.* slightly outperformed the other GAs.

Table 4.4: Comparison of the non-ALPS based GA with Thangiah *et al.* [52].

Instance	Thangiah *et al.* [52]	GA	Difference in Vehicles	% difference in Distance
P01	**591.73[10]**	613.76[10]	0	3.72
P02	**483.15[5]**	485.73[5]	0	0.53
P03	**694.49[10]**	714.23[10]	0	2.84
P04	1062.38[15]	**1044.50[15]**	0	-1.68
P05	**754.84[8]**	774[8]	0	2.54
P06	976.02[15]	**904.06[15]**	0	-7.96
P07	976.48[15]	**966.42[15]**	0	-1.04
Best solution	4/7	3/7	=7/7	-3/7,+4/7

Table 4.5: Comparison of the non-ALPS based GA with Ombuki *et al.* [44].

Instance	Ombuki *et al.* [44]	GA	Difference in Vehicle	% difference in Distance
P01	622.18[10]	**613.76[10]**	0	-1.37
P02	480.04[6]	485.73[5]	-1	
P03	**706.88[10]**	714.23[10]	0	1.04
P04	**1024.78[15]**	1044.50[15]	0	1.98
P05	785.15[8]	**774[8]**	0	-1.44
P06	908.88[15]	**904.06[15]**	0	-0.53
P07	918.05[16]	966.42[15]	-1	
Best solution	2/7	3/7	-2/7,=5/7	-3/7,+2/7

Table 4.6: Comparison of the non-ALPS based GA with the two well-known GA. Thangiah *et al.* [52] , Ombuki *et al.* [44]

Instance	Thangiah *et al.* [52]	Ombuki *et al.* [44]	GA
P01	**591.73[10]**	622.18[10]	613.76[10]
P02	483.15[5]	480.04[6]	485.73[5]
P03	**694.49[10]**	706.88[10]	714.23[10]
P04	1062.38[15]	**1024.78[15]**	1044.50[15]
P05	**754.84[8]**	785.15[8]	774[8]
P06	976.02[15]	908.88[15]	**904.06[15]**
P07	976.48[15]	918.05[16]	966.42[15]
Best Solution	3/7	1/7	1/7

4.2.3 Comparison between non-ALPS based GA and ALPS-GA

From Table 4.7 it could be deduced that the ALPS-GA performed better than the non-ALPS based GA in 5/7 instances with the highest minimized percentage of 3.71% and had the same results in 2 instances. Again this is considering the best results in 30 runs. However the corresponding averages are given by columns Avg(GA) and Avg(ALPS-GA) where it is shown that ALPS-GA outperformed the non-ALPS based GA.

Based on this result we used the ALPS-GA for the remaining experiments though the t-test result shown in Figure B.1 in appendix tells there is no significant difference between the two at 5% significant level.

Table 4.7: Comparison of the non-ALPS based GA with ALPS-GA.

Instances	GA	ALPS-GA	%Difference in Distance	Avg(GA)	Avg(ALPS-GA)
P01	**613.76[10]**	**613.76[10]**	0	623.07[10.33]	618.850[10]
P02	485.73[5]	**476.7[5]**	-1.89	496.53[5.4]	516.352[5.2]
P03	714.23[10]	**695.69[10]**	-2.66	722.01[10.47]	717.522[10]
P04	1044.50[15]	**1031.57[15]**	-1.25	1082.68[15]	1076.698[15]
P05	**774[8]**	**774[8]**	0	788.023[8]	799.14[8]
P06	904.05[15]	**894.91[15]**	-1.02	934.96[15.1]	929.242[15]
P07	966.42[15]	**931.85[15]**	-3.71	943.05[15.87]	955.837[15.47]
Best solution	2/7	7/7	-5/7,=2/7		

4.2.4 Comparison among three inter-layer transfer strategies

Tables 4.8 and 4.9 show the results attained from using the (3) inter layer transfers. In these tables, we show the best out of (30) runs per instance. Strategy BIR outperforms the other two with a score of 19/23 followed by RTW with 13/23 and 2/11 for the BTR.

It is hypothesised that BTR did not perform better because, the very best individual from the last layer at a point in time got replaced. This hypothesis is shown in Figure 4.1.

The figure shows that the algorithm easily finds the best known result for the P12 instance with BIR while BTR at a point increased in fitness. This contributed to the little increase in the results of some instances for BTR.

A single factor Anova test at 0.05 critical value among these three inter-layer transfer strategies is shown in Figure B.2 in Appendix B, which suggests no significant difference among the strategies.

Table 4.8: Comparison results for the three inter-layer transfer strategies in capacity constraint dataset.

Instances	BIR	RTW	BTR
P01	**613.76[10]**	**613.76[10]**	**613.76[10]**
P02	**476.7[5]**	**476.7[5]**	490.05[10]
P03	695.69[10]	**694.26[10]**	710.94[10]
P04	**1031.57[15]**	1043.3[15]	1057.9[15]
P05	**774[8]**	776.51[8]	798[8]
P06	**894.91[15]**	905.40[15]	908.63[15]
P07	**931.85[5]**	936.20[15]	955.66[15]
P12	**1318.95[8]**	**1318.95[8]**	**1318.95[8]**
P15	**2571.36[16]**	**2571.36[16]**	2582.11[16]
P18	**3859.37[24]**	3863.77[24]	3874.08[24]
P21	5831.3[36]	**5824.55[36]**	5855.64[36]
Best Solution	9/11	6/11	2/11

Table 4.9: Comparison results for two inter-layer transfer strategies in route and capacity constraints datasets

Instances	BIR	RTW
P08	**4993.50[25]**	5116.61[25]
P09	4464.27[25]	**4423.65[25]**
P10	4080.70[25]	**4072[25]**
P11	**3904.59[25]**	3999.78[25]
P13	**1318.95**[8]	**1318.95**[8]
P14	**1360.12**[8]	**1360.12**[8]
P16	**2575.33**[16]	**2575.33**[16]
P17	**2725.08**[16]	**2725.08**[16]
P19	**3867.61**[24]	3877.79[24]
P20	**4091.49**[24]	4097.06[24]
P22	**5848.55**[36]	5874.38[36]
P23	**6145.58**[36]	**6145.58**[36]
Best Solution	10/12	7/12

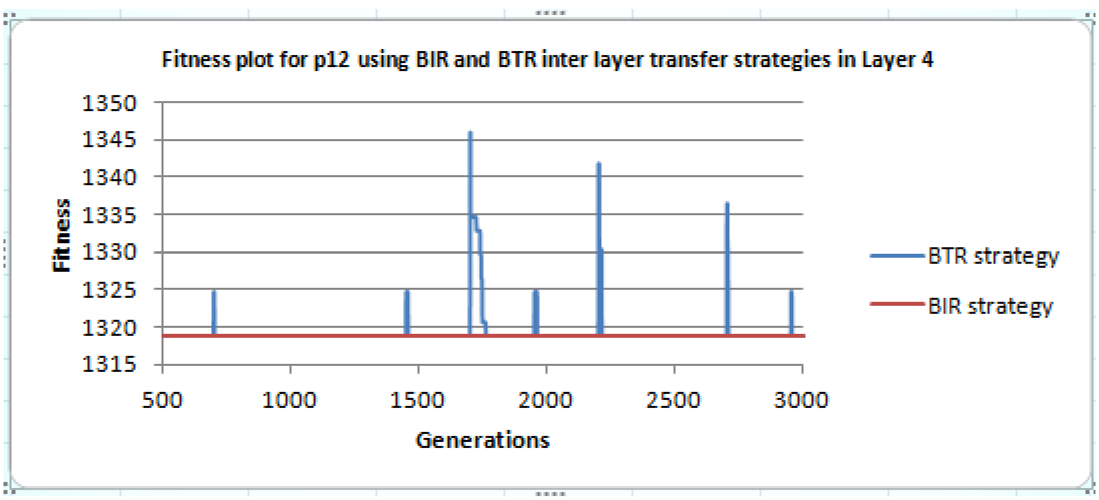

Figure 4.1: Fitness plot of layer 4 for P12 showing BTR and BIR. This shows that the best individual is occasionally replaced in BTR.

4.2.5 Comparing the weighted sum and the two multi-objective fitness evaluation strategies

We conducted the experiments on all the 23 instances using the same parameter settings in Table 4.3 and the different fitness evaluation strategies. We performed 30 runs per instance and showed the best results for the instances in Table 4.10 while the averages of the 30 runs are shown in Appendix C.

It is not so surprising to see normalised sum of ranks having the least number of best solution which is 6/23. This evaluation strategy had been designed for problems with more than 4 objectives where Pareto ranking happens to create some outliers. This notwithstanding we have seen from Table 4.10 that its results were not too worse than the best solutions determined.

Pareto ranking was second on the chart with a score of 10/23 which is 6 less than that of weighted sum. It is also not alarming to see two results for a dataset instance as these individuals are not better than each other. It signifies how one objective could be made better at the expense of the other. For example, the P10 instance recorded 4195.71 for 25 vehicles and 3920.60 for 26 vehicles as total distance. This also means that minimizing the number of vehicles comes with a shortfall of increasing the total cost. The weighted sum had a score of 16/23.

We show the pareto front for the P02 dataset instance in Figure 4.2. From this figure, all individuals with the same ranks are shown in series with the same shape and colour. The best ranks are the individuals close to the origin hence the purple diamond shape is the rank 1 individuals representing 5 vehicles and 476.7 as total distance. This analysis suggest that the three fitness evaluation strategies are suitable for solving the problem.

Table 4.10: Comparison results for the weighted sum with two multi-objective fitness evaluation strategies.

Instances	Weighted Sum[3.4.1]	Pareto[3.4.2]	Normalised[3.4.3]
P01	613.76[10]	613.76[10]	**612.14[10]**
P02	**476.7[5]**	**476.7[5]**	488.50[10]
P03	695.69[10]	**685.61[10]**	706.07[10]
P04	1031.57[15]	**1029.02[15]**	1049.62[15]
P05	774[8]	770.55[8]	**769.82[8]**
P06	**894.91[15]**	908.59[15]	899.93[15]
P07	**931.85[5]**	935.20[15]	934.44[15]
P08	**4993.50[25]**	5006.46[25]	5180.29[25]
P09	**4464.27[25]**	4591.15[25]	4563.88[25]
P10	4080.70[25]	4195.71[25],3920.60[26]	**4055.59[25]**
P11	**3904.59[25]**	3927.97[25],3820.71[26]	4033.62[25]
P12	**1318.95[8]**	**1318.95[8]**	**1318.95[8]**
P13	**1318.95[8]**	**1318.95[8]**	**1318.95[8]**
P14	**1360.12[8]**	**1360.12[8]**	1365.69[8]
P15	**2571.36[16]**	2576.08[16]	2583.23[16]
P16	**2575.33[16]**	**2575.33[16]**	2586.08[16]
P17	**2725.08[16]**	2731.30[16]	2731.30[16]
P18	3859.37[24]	**3840.35[24]**	3895.28[24]
P19	**3867.61[24]**	3872.64[24]	3914.05
P20	**4091.49[24]**	4097.06[24]	4097.06[24]
P21	5831.3[36]	**5820.69[36]**	5821.57[36]
P22	**5848.55[36]**	5874.98[36]	5970.49[36]
P23	**6145.58[36]**	**6145.58[36]**	**6145.58[36]**
Best Solution	16/23	10/23	6/23

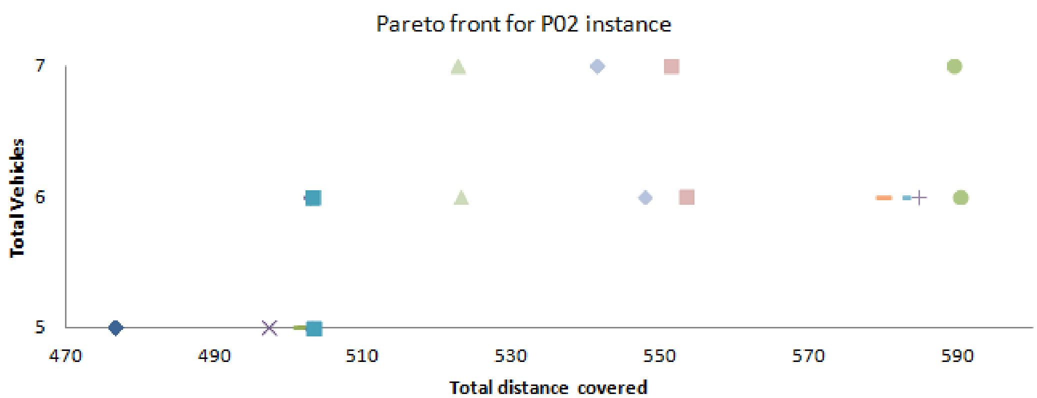

Figure 4.2: Pareto front plot for the P02 instance

4.2.6 Comparing ALPS-GA to known GAs (Weighted sum) using BIR inter-layer transfer.

We followed the comparison strategy used by both Thangiah *et al.* [52] and Ombuki *et al.* [44]. These comparisons can be found in Tables 4.11, 4.12, and 4.13. We compared the number of vehicles used as well as the percentage difference in distance which can be found in Tables 4.11, and 4.12. These values indicate the use of less, equal or more vehicles. It is simply explained as; negative(-) shows reduction, zero(0) indicates equal values and positive(+) denotes an increase in the vehicles used.

From Table 4.11, the proposed algorithm used the same number of vehicles in 17/23 instances in comparison with Thangiah *et al.* [52] and 19/23 when compared to Ombuki *et al.* [44] in Table 4.12. There were 3/23 and 4/23 reduction in vehicles compared to Thangiah *et al.* [52] and Ombuki *et al.* [44] respectively. This implies there was no increase when compared with Ombuki *et al.* [44] but increase in 3/23 instances compared to Thangiah *et al.* [52]. We can therefore say that the ALPS-GA is comparable to the Thangiah *et al.* [52] in terms of vehicles used. The reduction in 4/23 instances show that our algorithm outperformed that of Ombuki *et al.* [44] with respect to vehicles used.

The proposed algorithm can be said to have outperformed that of Thangiah *et al.* [52] since out of 17 instances with equal vehicles, ALPS-GA minimized 10/17 instances , with the highest reduction percentage of 9.06.

Out of the 19 instances compared with Ombuki *et al.* [44], there were reduction in 13/19 and 4/19 increase with 2/19 being equal. We add that, in both comparisons from Tables 4.11 and 4.12, we realised a decrease in both vehicles and distance for the P02 instance. This reduction in both objectives also happened in Table 4.11 for the P20 instance. Finally, from Table 4.13 (comparing the three (3) algotithms), ALPS-GA was in the lead with 9/23 followed by Ombuki *et al.* [44] with 5/23 and Thangiah [52] with 4/23. This showed that the proposed algorithm outperformed both algorithms.

We show the network and graphs generated from the ALPS-GA algorithm for P02 in Figures 4.3, 4.4, 4.5 while networks for P10, P14 and P23 are shown in the appendix C as well as their optimal routes in Figures C.3, C.4 , C.5. We also show results of the average of thirty (30) in Table C.1 in the appendix C.

Table 4.11: Comparison of the ALPS-GA with Thangiah *et al.* [52] (Weighted sum).

Instances	Thangiah *et al.* [52]	ALPS-GA	Difference in Vehicle	% difference Distance
P01	**591.73[10]**	613.76[10]	0	3.73
P02	483.15[5]	**476.7[5]**	0	-1.35
P03	**694.49[10]**	695.69[10]	0	0.17
P04	1062.38[15]	**1031.57[15]**	0	-2.99
P05	**754.84[8]**	774[8]	0	2.54
P06	976.02[15]	**894.91[15]**	0	-9.06
P07	976.48[15]	**931.85[15]**	0	-4.79
P08	**4812.52[25]**	4993.50[25]	0	3.76
P09	**4284.62[25]**	4464.27[25]	0	4.19
P10	4291.45[25]	**4080.70[25]**	0	-5.16
P11	4092.68[25]	**3904.59[25]**	0	-3.73
P12	1421.94[8]	**1318.95[8]**	0	-7.81
P13	**1318.95[8]**	**1318.95[8]**	0	0
P14	**1360.12[8]**	**1360.12[8]**	0	0
P15	3059.15[15]	2571.36[16]	1	
P16	2719.98[16]	**2575.33[16]**	0	-5.62
P17	2894.69[16]	**2725.80[16]**	0	-6.19
P18	5462.90[22]	3859.37[24]	2	
P19	3956.61[24]	**3867.61[24]**	0	-2.3
P20	4344.81[27]	**4091.49[24]**	-3	
P21	6872.11[34]	5831.3[36]	2	
P22	5985.32[37]	**5848.55[36]**	-1	
P23	6288.04[39]	**6145.58[36]**	-3	
Best Found	7/23	15/23	-3/23, +3/23, =17/23	-10/17, +5/17

Table 4.12: Comparison of the ALPS-GA with known Ombuki *et al.* [44] (Weighted sum)

Instances	Ombuki *et al.* [44]	ALPS-GA	Difference in Vehicle	% difference Distance
P01	622.18[10]	**613.76[10]**	0	-1.37
P02	480.04[6]	**476.7[5]**	-1	
P03	706.88[10]	**695.69[10]**	0	-1.61
P04	**1024.78[15]**	1031.57[15]	0	0.66
P05	785.15[8]	**774[8]**	0	-1.44
P06	908.88[15]	**894.91[15]**	0	-1.56
P07	918.05[16]	931.85[15]	-1	
P08	**4690.18[25]**	4993.50[25]	0	6.47
P09	**4240.08[25]**	4464.27[25]	0	5.29
P10	3984.78[26]	4080.70[25]	-1	
P11	**3880.65[25]**	3904.59[25]	0	0.62
P12	**1318.95[8]**	**1318.95[8]**	0	0
P13	**1318.95[8]**	**1318.95[8]**	0	0
P14	1365.69[8]	**1360.12[8]**	0	-0.37
P15	2579.25[16]	**2571.36[16]**	0	-0.31
P16	2587.87[16]	**2575.33[16]**	0	-0.49
P17	2731.37[16]	**2725.80[16]**	0	-0.21
P18	3903.85[24]	**3859.37[24]**	0	-1.74
P19	3900.61[24]	**3867.61[24]**	0	-0.85
P20	4097.06[24]	**4091.49[24]**	0	-0.14
P21	5926.49[36]	**5831.3[36]**	0	-1.22
P22	5913.59[36]	**5848.55[36]**	0	-1.28
P23	6145.58[37]	6145.58[36]	-1	
Best Found	6/23	16/23	-4/23, =19/23	-13/19, +4/19

Table 4.13: Comparison of the ALPS-GA with the known GAs (Weighted sum).

Instances	Thangiah *et al.* [52]	Ombuki *et al.* [44]	ALPS-GA
P01	**591.73[10]**	622.18[10]	613.76[10]
P02	483.15[5]	480.04[6]	**476.7[5]**
P03	**694.49[10]**	706.88[10]	695.69[10]
P04	1062.38[15]	**1024.78[15]**	1031.57[15]
P05	**754.84[8]**	785.15[8]	774[8]
P06	976.02[15]	908.88[15]	**894.91[15]**
P07	976.48[15]	918.05[16]	931.85[15]
P08	4812.52[25]	**4690.18[25]**	4993.50[25]
P09	4284.62[25]	**4240.08[25]**	4464.27[25]
P10	4291.45[25]	3984.78[26]	4080.70[25]
P11	4092.68[25]	**3880.65[25]**	3904.59[25]
P12	1421.94[8]	**1318.95[8]**	**1318.95[8]**
P13	**1318.95[8]**	**1318.95[8]**	**1318.95[8]**
P14	**1360.12[8]**	1365.69[8]	**1360.12[8]**
P15	3059.15[15]	2579.25[16]	2571.36[16]
P16	2719.98[16]	2587.87[16]	**2575.33[16]**
P17	2894.69[16]	2731.37[16]	**2725.80[16]**
P18	5462.90[22]	3903.85[24]	3859.37[24]
P19	3956.61[24]	3900.61[24]	**3867.61[24]**
P20	4344.81[27]	4097.06[24]	**4091.49[24]**
P21	6872.11[34]	5926.49[36]	5831.3[36]
P22	5985.32[37]	5913.59[36]	**5848.55[36]**
P23	6288.04[39]	6145.58[37]	6145.58[36]
Best Found	5/23	6/23	10/23

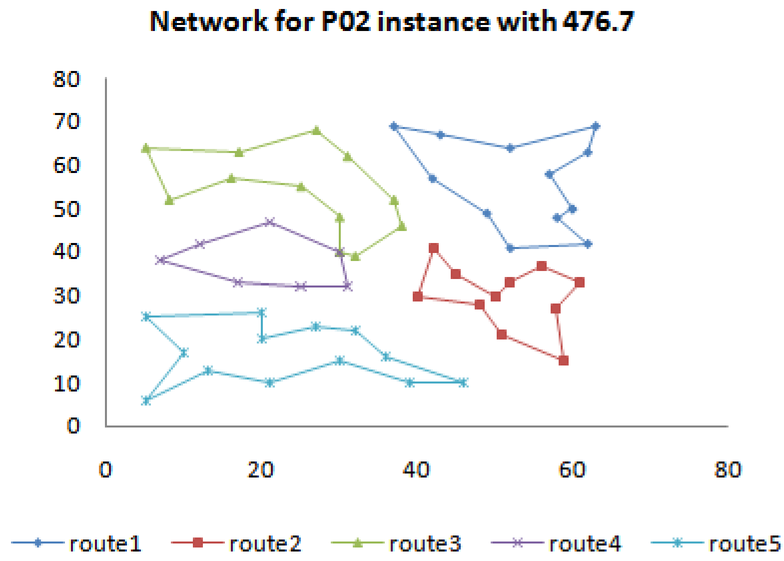

Figure 4.3: Generated Network for P02.

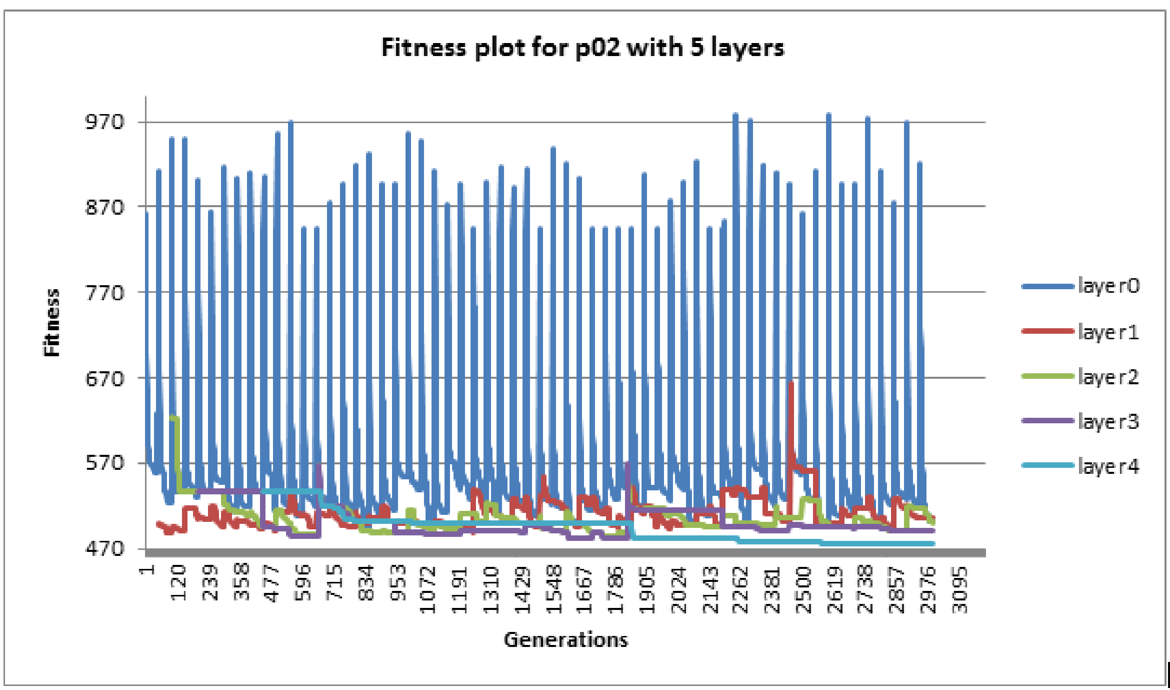

Figure 4.4: Fitness plot for P02 with 5 layers.

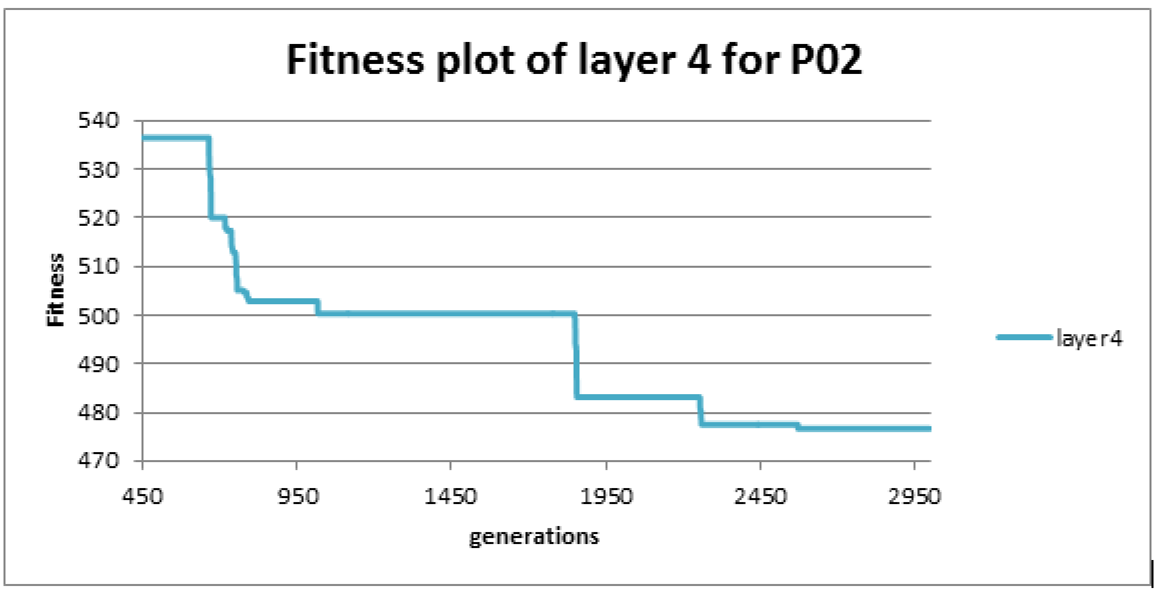

Figure 4.5: Fitness plot for P02 in layer4.

Explanation of the network and fitness plots

The network in Figure 4.3 shows 50 customers with 4 depots. This figure shows that 3/4 depots had one vehicle (route) each to serve its customers. The last depot however used two vehicles to serve the customers and obeyed the constraints that each customer should be visited exactly once with the vehicle departing from and arriving at the same depot.

In Figure 4.4, there are five layers with four showing spiky fitness curves. The process of inter layer transfer caused the spikes in the fitness plot. Also the consistent rapid oscillation of layer 0's fitness curve is as a result of introducing new individuals at regular intervals. The transfer of individual at each layer except layer 4 seems to happen differently and it so occurs due to different age limits at the individual layers. Figure 4.5 illustrates the last layer (in the fitness plot).

The fitness of the best individuals is seen to have started from the 450th generation as the last layer becomes active at this generation during the evolutionary process, it maintained the fitness value for a short while and started the minimization process. Though the slope was not steadily smooth, it shows that the fitness kept improving with intermediate convergence. An example of the intermediate convergence can be seen from the 700th to 955th generations thus, wherever the curve flattens out.

4.2.7 Comparing ALPS-GA to known non-GAs (Weighted sum).

Similar to the format used in comparing ALPS-GA to the known GAs, we compare ALPS-GA to known non-GAs .

V(CGW) and V(RBL) provides the vehicle difference between ALPS-GA and the CGW [13], RBL [47] algorithms respectively. The negative values in these columns indicate the reduction in the number of vehicles used, while positive denote increase, with zero(0) meaning equal number of vehicles used. The equal numbers are highly preferred for the comparison though positive and negative have their own analytical implications.
ALPS-GA proved to be comparable to both algorithms in terms of vehicles used (can even be concluded that it performed better than CGW [13]) as it provided a reduction in 15/23 instances, 8/23 equal values for the remaining instances and no increment. In a similar way, there was only one positive value when compared with RBL [47], and 6/23 reduction while the remaining 16 were of equal values.

Now with the fair comparison based on equal values for vehicles, it is not surprising that there was no bolded values under the CGW, which also confirm that ALPS-GA was better than it. However on the other hand, a win could be awarded to RBL as it has 13/23 instances with better results while ALPS-GA had 5/23.
The highest depreciation percentage in favour of ALPS-GA was -3.33 from CGW and -0.41 from RBL.

Table 4.14: Comparison of ALPS-GA with known non GAs

Instances	CGW	RBL	ALPS-GA	$V_{(CGW)}$	$V_{(RBL)}$	$D_{(CGW)}$	D_{RBL}
P01	582.3[11]	576.86[11]	613.76[10]	-1	-1		
P02	476.7[5]	**473.53[5]**	476.7[5]	0	0		0.67
P03	641.2[11]	641.2[11]	695.69[10]	-1	-1		
P04	1026.9[16]	**1003.87[15]**	1031.57[15]	-1	0		
P05	756.6[8]	**750.26[8]**	774[8]	0	0	3.16	3.23
P06	883.6[16]	876.50[16]	894.91[15]	-1	-1		
P07	898.5[17]	**892.58[15]**	931.85[15]	-2	0		4.4
P08	4511.6[27]	**4485.09[25]**	4993.50[25]	-2	0		11.3
P09	3950.9[26]	3937.82[26]	4464.27[25]	-1	-1		
P10	3815.6[28]	3669.38[26]	4080.70[25]	-3	-1		
P11	3733.0[27]	3648.95[26]	3904.59[25]	-2	-1		
P12	1327.3[8]	**1318.95[8]**	**1318.95[8]**	0	0	-0.63	0
P13	1345.9[8]	**1318.95[8]**	**1318.95[8]**	0	0	0	0
P14	1372.5[8]	1365.69[8]	**1360.12[8]**	0	0	-0.91	-0.41
P15	2610.3[16]	**2551.46[16]**	2571.36[16]	0	0	-1.51	0.78
P16	2605.7[16]	**2572.23[16]**	2575.33[16]	0	0	-1.18	0.12
P17	2816.6[18]	2731.37[16]	**2725.80[16]**	-2	0	-3.33	-0.20
P18	3877.4[25]	**3781.04[23]**	3859.37[24]	-1	1		
P19	3864.0[24]	**3827.06[24]**	3867.61[24]	0	0	0.09	1.06
P20	4272.0[28]	4097.06[24]	**4091.49[24]**	-4	0		-0.14
P21	5791.5[37]	**5656.47[36]**	5831.3[36]	-1	0		3.49
P22	5857.4[37]	**5718.00[36]**	5848.55[36]	-1	0		2.3
P23	6494.6[41]	6145.8[36]	6145.8[36]	-5	0		0
Best Solution	0/23	13/23	5/23	-15/23	-6/23		0

4.2.8 Comparison of ALPS-GA to known GAs using Normalised Sum of Ranks

We compare our results to Thangiah *et al.* [52] and Ombuki *et al.* [44] where Ombuki *et al.* [44] used pareto ranking.
From Table 4.17, the ALPS-GA had the same number of vehicles compared to Thangiah *et al.* [52] but proved more superior when compared to Ombuki *et al.* [44]. It had reduction in 9 instances and the same number of vehicles in 17 instances. Two solutions can be observed under Ombuki *et al.*'s [44] column which resulted from the fact that none of the two solutions dominated the other.

From Tables 4.15 and 4.16, ALPS-GA performs better than the two algorithms with 11/23 and 12/23 respectively while recording the reduction percentages of 8.49 and 4.81. When all three algorithms are compared together, we can see from the Table 4.17 that they are comparable although the proposed algorithm had 9/23 followed by Ombuki *et al.* [44] with 7/23 and lastly 5/23 for the Thangiah *et al.* [52]. The averages of 30 runs for each instance is given in Appendix C (C.2)

Table 4.15: Comparison of ALPS-GA to known Thangiah [52] using normalised sum of ranks

Inst.	Thangiah *et al.* [52]	ALPS-GA	Difference in Vehicle	% difference Distance
P01	**591.73[10]**	612.14[10]	0	3.45
P02	**483.15[5]**	488.50[5]	0	1.11
P03	**694.49[10]**	706.07[10]	0	1.67
P04	1062.38[15]	**1049.62[15]**	0	-1.22
P05	**754.58[8]**	769.82[8]	0	2.02
P06	976.02[15]	**899.63[15]**	0	-8.49
P07	976.48[15]	**934.44[15]**	0	-5.81
P08	**4812.52[25]**	5180.29[25]	0	7.64
P09	**4284.62[25]**	4563.88[25]	0	6.52
P10	4291.45[25]	**4055.59[25]**	0	-5.82
P11	4092.68[25]	**4033.62[25]**	0	-1.46
P12	1421.94[8]	**1318.95[8]**	0	-7.81
P13	**1318.95[8]**	**1318.95[8]**	0	
P14	**1360.12[8]**	1365.69[8]	0	0.41
P15	3059.15[15]	2583.23[16]	1	
P16	2719.98[16]	**2586.08[16]**	0	-5.12
P17	2894.69[16]	**2731.37[16]**	0	-5.98
P18	5462.90[22]	3895.28[24]	2	
P19	3956.61[24]	**3914.05[24]**	0	-1.09
P20	4344.81[27]	**4097.06[24]**	-3	
P21	6872.11[34]	5821.57[36]	2	
P22	5985.33[37]	5970.49[36]	-1	
P23	6288.04[39]	6145.58[36]	-3	
Best Solution	8/23	11/23	-3/23,+3/23,=17/23	-9/17,7/17,=1/17

Table 4.16: Comparison of ALPS-GA to known Ombuki *et al.* [44] using normalised sum of ranks.

Inst.	Ombuki *et al.* [44]	ALPS-GA	Difference in Vehicle	% difference Distance
P01	600.63[11]	612.14[10]	-1	-
P02	480.04[6]	488.50[5]	-1	
P03	683.15[11]	706.07[10]	-1	
P04	**1034.59[15]**	1049.62[15]	0	1.45
P05	778.01[8]	**769.82[8]**	0	-1.06
P06	916.71[15],900.44[16]	**899.63[15]**	0,-1	-1.90
P07	922.83[16]	934.44[15]	-1	
P08	**4672.56[25]**	5180.29[25]	0	10.87
P09	**4332.32[25]**,4243.74[26]	4563.88[25]	0,-1	5.34
P10	3953.24[26]	4055.59[25]	-1	
P11	**3962.17[25]**,3876.26[26]	4033.62[25]	0,-1	1.80
P12	**1318.95[8]**	**1318.95[8]**	0	
P13	**1318.95[8]**	**1318.95[8]**	0	
P14	**1365.69[8]**	**1365.69[8]**	0	
P15	**2579.25[16]**	2583.23[16]	0	
P16	2596.83[16]	**2586.08[16]**	0	-0.42
P17	**2731.37[16]**	**2731.37[16]**	0	
P18	3897.22[24]	**3895.28[24]**	0	-0.05
P19	3972.80[24]	**3914.05[24]**	0	-1.50
P20	**4097.06[24]**	**4097.06[24]**	0	
P21	6101.68[36]	**5821.57[36]**	0	-4.81
P22	5984.87[36]	**5970.49[36]**	0	-0.24
P23	6145.35[37]	6145.58[36]	-1	
Best Solution	10/23	12/23	-9/23,=17/23	-7/17,+5/17,=5/17

Table 4.17: Comparison of ALPS-GA to known GAs using normalised sum of ranks.

Inst.	Thangiah *et al.* [52]	Ombuki *et al.* [44]	ALPS-GA
P01	**591.73[10]**	600.63[11]	612.14[10]
P02	483.15[5]	480.04[6]	488.50[5]
P03	694.49[10]	683.15[11]	706.07[10]
P04	1062.38[15]	**1034.59[15]**	1049.62[15]
P05	**754.58[8]**	778.01[8]	769.82[8]
P06	976.02[15]	916.71[15],900.44[16]	**899.63[15]**
P07	976.48[15]	922.83[16]	934.44[15]
P08	4812.52[25]	**4672.56[25]**	5180.29[25]
P09	**4284.62[25]**	4332.32[25],4243.74[26]	4563.88[25]
P10	4291.45[25]	3953.24[26]	4055.59[25]
P11	4092.68[25]	**3962.17[25]**,3876.26[26]	4033.62[25]
P12	1421.94[8]	**1318.95[8]**	**1318.95[8]**
P13	**1318.95[8]**	**1318.95[8]**	**1318.95[8]**
P14	**1360.12[8]**	1365.69[8]	1365.69[8]
P15	3059.15[15]	2579.25[16]	2583.23[16]
P16	2719.98[16]	2596.83[16]	**2586.08[16]**
P17	2894.69[16]	**2731.37[16]**	**2731.37[16]**
P18	5462.90[22]	3897.22[24]	3895.28[24]
P19	3956.61[24]	3972.80[24]	**3914.05[24]**
P20	4344.81[27]	**4097.06[24]**	**4097.06[24]**
P21	6872.11[34]	6101.68[36]	**5821.57[36]**
P22	5985.33[37]	5984.87[36]	**5970.49[36]**
P23	6288.04[39]	6145.35[37]	6145.58[36]
Best Solution	5/23	7/23	9/23

4.2.9 Comparing ALPS-GA using Pareto ranking with that of Ombuki et al. [44].

ALPS-GA has once again shown its efficiency as it reduces 17/23 compared to 6/23 for the Ombuki *et al.* [44]. The reduction percentage ranges from 0.12 to 4.83. These details can be viewed from Table 4.18.

Table 4.18: Comparison of ALPS-GA to Ombuki *et al.* [44] with both using Pareto Ranking.

Inst.	Ombuki *et al.* [44]	ALPS-GA	Difference in Vehicle	% difference Distance
P01	600.63[11]	613.76[10]	-1	-
P02	480.04[6]	**476.7[5]**	-1	
P03	683.15[11]	685.61[10]	-1	
P04	1034.59[15]	**1029.02[15]**	0	-0.54
P05	778.01[8]	**770.55[8]**	0	-0.97
P06	916.71[15],900.44[16]	**908.59[15]**	0,-1	-0.89
P07	922.83[16]	935.20[15]	-1	
P08	**4672.56[25]**	5006.46[25]	0	7.15
P09	**4332.32[25]**,4243.74[26]	4591.15[25]	0,-1	5.97
P10	3953.24[26]	4195.71[26],**3920.60[25]**	-1	
P11	3962.17[25],3876.26[26]	**3927.97[25],3820.71[26]**	0	-0.87,-1.45
P12	**1318.95[8]**	**1318.95[8]**	0	
P13	**1318.95[8]**	**1318.95[8]**	0	
P14	1365.69[8]	**1360.12[8]**	0	
P15	2579.25[16]	**2576.08[16]**	0	-0.12
P16	2596.83[16]	**2575.33[16]**	0	-0.42
P17	**2731.37[16]**	**2731.37[16]**	0	
P18	3897.22[24]	**3840.35[24]**	0	-1.48
P19	3972.80[24]	**3872.64[24]**	0	-2.57
P20	**4097.06[24]**	**4097.06[24]**	0	
P21	6101.68[36]	**5820.69[36]**	0	-4.83
P22	5984.87[36]	**5874.98[36]**	0	-1.87
P23	6145.35[37]	6145.58[36]	-1	
Best Solution	6/23	17/23	-9/23,=17/23	-7/17,+5/17,=5/17

4.2.10 Comparison of the ALPS-GA to known Algorithms compared in [38] in addition to the nomadic algorithm from [11].

This comparison is performed following Equation 4.1 which calculates the total distance covered by the available routes.

$$Fitness = \sum_{d=1}^{D} Dist \tag{4.1}$$

where D is the number of depots, Dist is the total distance travelled at a depot d.

The algorithms used for this comparison are well explained in [38]. As can be seen from Table 4.19, while all the GA variants could not outperform the other algorithms, ALPS-GA fairly performed better than all GAs but nomadic GA algorithm (GA4) [11] as both detected 1/5 best solutions. Despite ALPS-GA can be said to be better than known GAs, it is somehow less optimal compared to other methods. We show the network for P01 instance along with its fitness plot in Figures 4.6 and 4.7 . Lastly we show the customers in the various routes with their distances travelled for the results obtained for P01 instance in Table 4.20

Table 4.19: Comparison results for existing algorithms with ALPS-GA using Equation 4.1

Instances	P01	P02	P03	P04	P06	score
EXACT	576.87	473.53	641.15	1001.04	876.5	5/5
FIND	576.86	473.53	641.18	1003.86	876.5	4/5
ACO	576.86	484.28	645.16	1020.52	878.34	1/5
ITS	576.87	473.53	641.19	1001.04	876.5	5/5
ALNS	576.87	473.53	641.19	1001.4	876.7	5/5
CGL	576.86	473.87	645.15	1006.66	877.84	2/5
PIACO	576.86	473.35	641.18	1001.49	876.5	5/5
ACO-WM	576.86	473.53	641.18	1001.49	876.5	5/5
THANGIAH *et al.*	591.73	483.15	694.49	1062.38	976.02	0/5
OMBUKI *et al.*	622.04	480.04	706.88	1024.78	908.88	0/5
GA3	598.45	478.75	699.23	1011.36	882.48	0/5
GA4	580.85	473.53	680.2	1010.25	878.88	1/5
ALPS-GA	576.87	476.7	648.76	1028	888.75	1/5

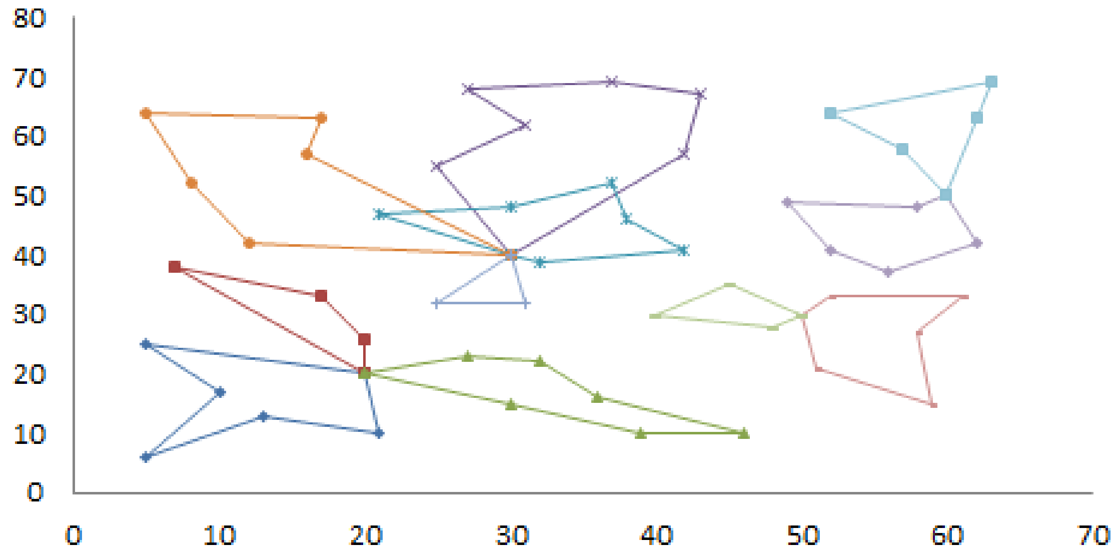

Figure 4.6: Network for P01 with 576.87 as distance

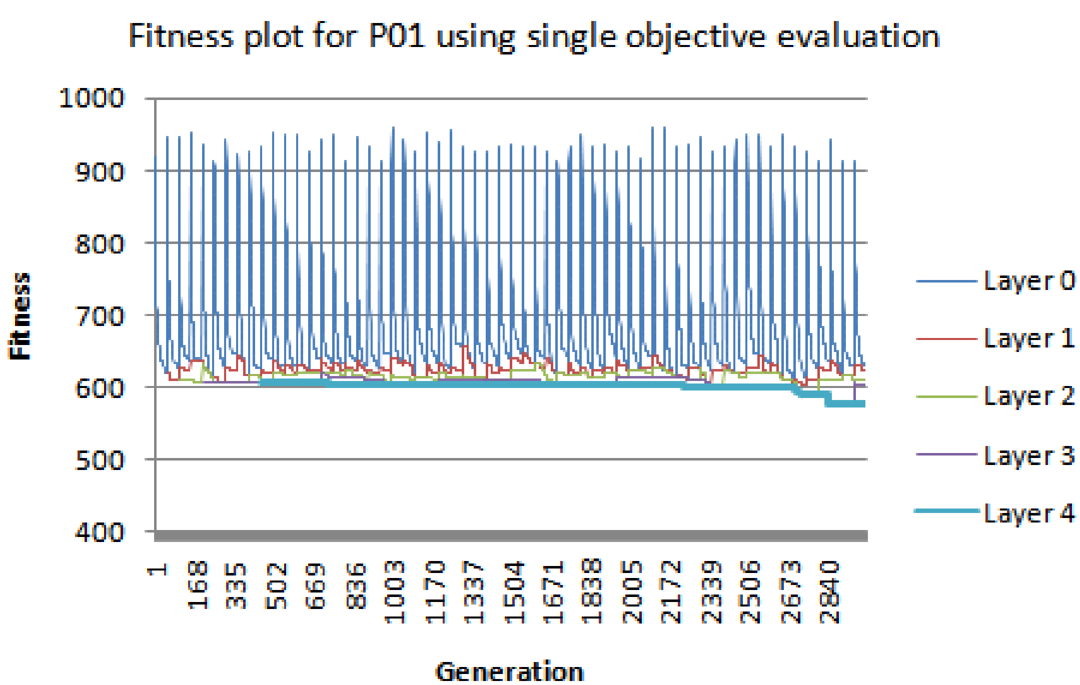

Figure 4.7: Fitness plot for P01 using Equation 4.1

Table 4.20: The optimised route from ALPS-GA for P01 using Equation 4.1

Routes	Ordered customers in route	Distance
1	51-13-41-40-19-42-51	66.552
2	51-4-18-25-51	46.9997
3	51- 17-37-15-33-45-44-51	60.064
4	52-48-8-26-31-28-22-52	77.455
5	52-46-11-32-1-27-6-52	53.439
6	52-14-24-43-7-23-52	81.397
7	52-12-47-52	23.496
8	53-9-34-30-39-10-53	50.411
9	53-38-5-49-53	25.217
10	54-29-2-16-50-21-54	41.086
11	54-20-3-36-35-54	47.673
	total distance	576.87

To end this chapter, we pictorially illustrate the processes of having the geographically dispersed customers, clustered customers to their nearest depots and finally the optimised routes formed after the evolutionary process. We used the P02 instance for this illustration. These are shown by Figures 4.8, 4.9, and 4.10 respectively. Table 4.21 shows the customers in a route at the individual depots with their corresponding distances travelled in each of the routes.

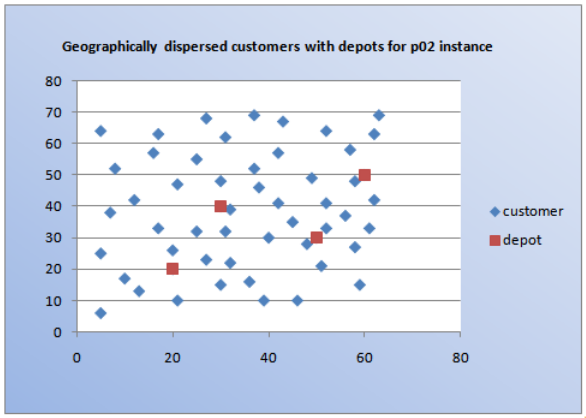

Figure 4.8: Geographically dispersed customers and depots.

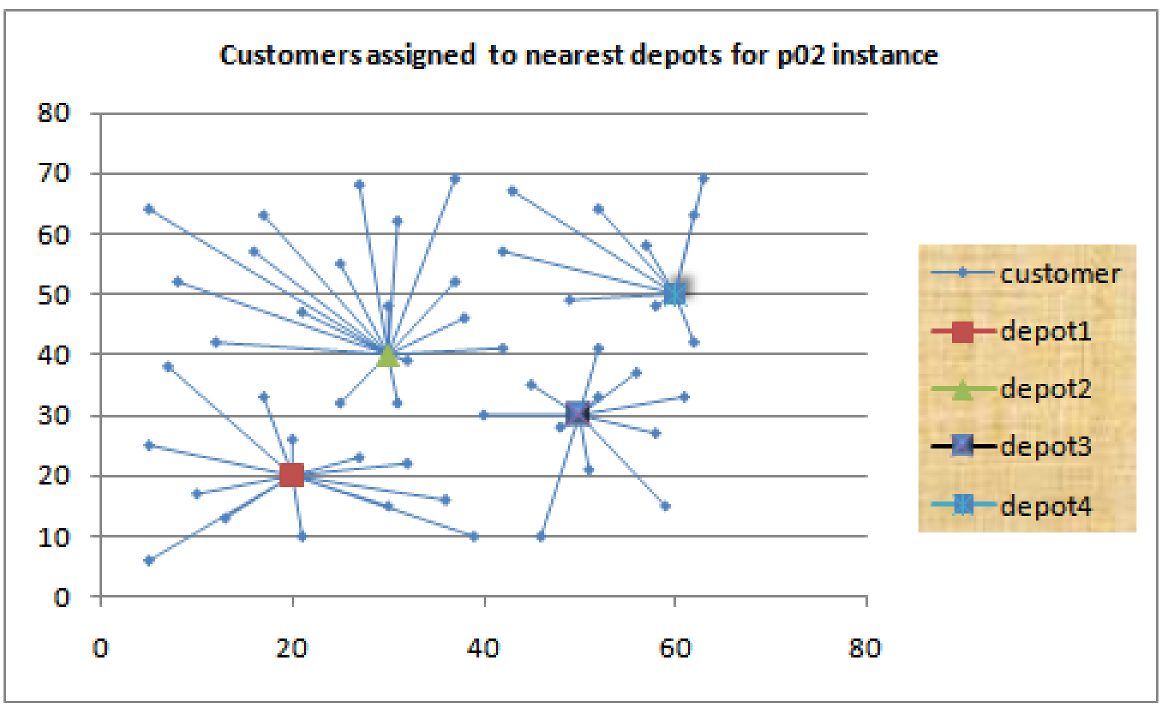

Figure 4.9: Customers assigned to their nearest depots

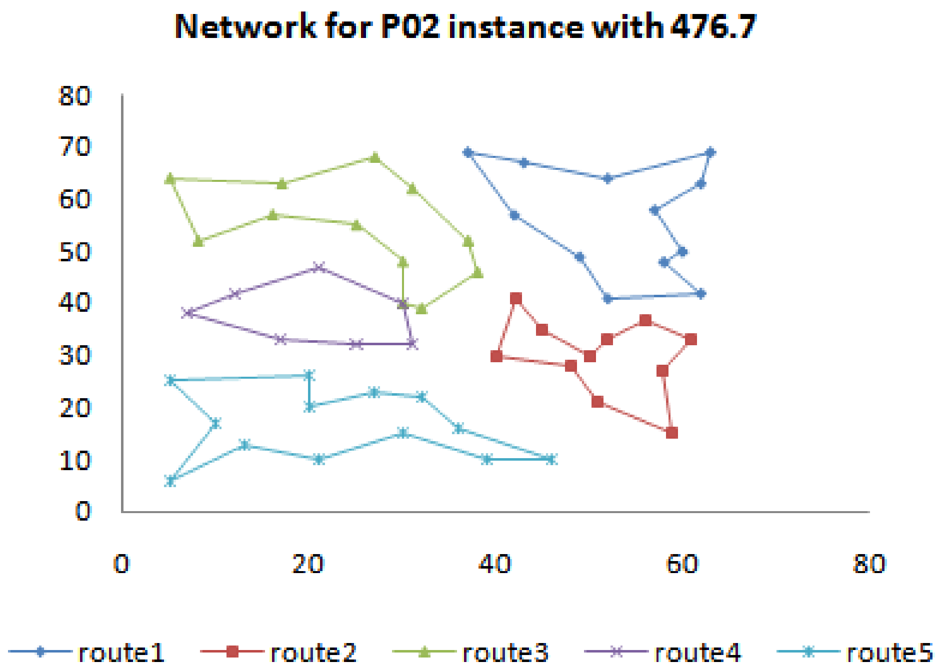

Figure 4.10: network for p02 instance showing 4depots and 5 routes

Table 4.21: The optimised route from ALPS-GA for P02 dataset

Routes	Ordered customers in route	Distance
1	51-17-37-15-33-45-44-42-19-40-41-13-4-51	120.904
2	52-6-14-25-18-47-12-52	61.405
3	52-46-32-1-8-26-7-43-24-23-48-27-52	107.258
4	53-9-50-34-30-39-10-49-5-11-38-53	83.867
5	54-20-35-36-3-28-31-22-2-16-21-29-54	101.856
	total distance	476.7

Chapter 5

Conclusion and Future Works

The study of vehicle routing problem (VRP) has benefited several industries like the distribution, logistics, and supply chain. The intractability of the problem is experienced when a customer is added to the already existing routes. This addition increases the problem exponentially. The NP-hard MDVRP (fixed destination) is studied in this thesis. The focus was on instances with either capacity constraint or both capacity and route length constraints collated by Cordeau [57]. There has been a number of GAs applied to this problem in literature and to the knowledge of the authors, there is one paper that focused on the use of two multi-objective fitness evaluation strategies. Little research, considered the MDVRP with a multi-population approach but without the multi-objective evaluation strategies. This thesis is the first multi-population multi-objective approach to consider for the MDVRP.

The Age Layered Population Structure (ALPS) based GA is implemented with three inter-layer transfer strategies. The analysis drawn was, the type of inter layer transfer affects the overall results. The mutation employed was derived from merging the approaches of two existing mutations in literature.

There was an additional multi-objective fitness evaluation strategy (normalized sum of ranks) to the already used Pareto ranking from [44] to solve the MDVRP. The experimental results are competitive to the compared results with improvements in some instances.

Lastly, result from computing only distance travelled suggest that the multi-population GA approach to the problem outperforms that of the single population approaches in literature.

Below are some considerations that can be made for future work:

1. The proposed ALPS-GA could also be enhanced with local search to improve the results.
2. Other fitness evaluation strategies could be tried on the problem.
3. The proposal of new inter-layer transfer for ALPS is worth consideration.
4. There are several clustering methods in literature other than the one implemented he implementation of another clustering method is plausible.
5. ALPS-GA could be tried on other optimization problems especially the MDVRP with time windows.

Bibliography

[1] Awuley, Anthony. *"Feature Selection and Classification Using Age Layered Population Structure Genetic Programming."* (2015).

[2] Baldacci, Roberto, and Aristide Mingozzi. *"A unified exact method for solving different classes of vehicle routing problems."*, Mathematical Programming 120, no. 2 (2009): 347-380.

[3] Ball, Michael O., B. L. Golden, A. A. Assad, and L. D. Bodin. *"Planning for truck fleet size in the presence of a commoncarrier option."* Decision Sciences 14, no. 1 (1983): 103-120.

[4] Banzhaf, Wolfgang, Peter Nordin, Robert E. Keller, and Frank D. Francone. *"Genetic programming"* : an introduction. Vol. 270. San Francisco: Morgan Kaufmann, 1998.

[5] Bentley, Peter J., and Jonathan P. Wakefield. *"Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms."* In Soft computing in engineering design and manufacturing, pp. 231-240. Springer London, 1998.

[6] Bergen, Steven, and Brian J. Ross. *"Evolutionary art using summed multi-objective ranks."* In Genetic Programming Theory and Practice VIII, pp. 227-244. Springer New York, 2011.

[7] Bongard, Josh C., and Gregory S. Hornby. *"Guarding against premature convergence while accelerating evolutionary search."* In Proceedings of the 12th annual conference on Genetic and evolutionary computation, pp. 111-118. ACM, 2010.

[8] Branke, Jrgen, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowiski, eds. *"Multiobjective optimization: interactive and evolutionary approaches."* Vol. 5252. Springer, 2008.

[9] Budin, Leo, Marin Golub, and Andrea Budin. "*Traditional techniques of genetic algorithms applied to floating-point chromosome representations.*", sign 1, no. 11 (2010): 52.

[10] Chand, Padmabati, and J. R. Mohanty. *"Multi objective genetic approach for solving vehicle routing problem."* International Journal of Computer Theory and Engineering 5, no. 6 (2013): 846.

[11] Chandrabhushan Prasad, S. Siva Sathya. *"A Nomadic Genetic Algorithm Approach with GMOUX Crossover for Multi Depot Vehicle Routing Problem"*, International Journal of Advanced Research in Computer Science and Software Engineering Volume 4 , Issue 5,May 2014 ISSN: 2277 128X

[12] Chang, Yaw, and Lin Chen. *"Solve the vehicle routing problem with time windows via a genetic algorithm."*, Discrete Continuous Dynamical Systems Supplement (2007): 240-249.

[13] Chao, I-Ming, Bruce L. Golden, and Edward Wasil. *"A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions."* American Journal of Mathematical and Management Sciences 13, no. 3-4 (1993): 371-406.

[14] Contardo, Claudio, and Rafael Martinelli. *"A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints."*, Discrete Optimization 12 (2014): 129-146.

[15] Cordeau, Jean-Franois, Michel Gendreau, and Gilbert Laporte. *"A tabu search heuristic for periodic and multi-depot vehicle routing problems."* Networks 30, no. 2 (1997): 105-119.

[16] Crevier, Benoit, Jean-Franois Cordeau, and Gilbert Laporte. *"The multi-depot vehicle routing problem with inter-depot routes."* European Journal of Operational Research 176, no. 2 (2007): 756-773.

[17] Dantzig, George Bernard; Ramser, John Hubert (October 1959). *"The Truck Dispatching Problem"* (PDF). Management Science. 6 (1): 8091. doi:10.1287/mnsc.6.1.80.

[18] Deb, Kalyanmoy. *"Multi-objective optimization using evolutionary algorithms."* Vol. 16. John Wiley & Sons, 2001.

[19] Desrosiers, Jacques, Yvan Dumas, Marius M. Solomon, and Franois Soumis. *"Time constrained routing and scheduling."*, Handbooks in operations research and management science 8 (1995): 35-139.

[20] Dommers, Sander, et al. *"A PTAS for the multiple depot vehicle routing problem."* TU/e, Technische Universiteit Eindhoven, Department of Mathematics and Computing Science, 2008.

[21] Ehrgott, Matthias. *"Multicriteria optimization".* Springer Science & Business Media, 2006.

[22] El-Sherbeny, Nasser A. *"Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods."*, Journal of King Saud University-Science 22, no. 3 (2010): 123-131.

[23] G. Clarke, J.W. Wright, *"Scheduling of vehicles from a central depot to a number of delivery points",* Oper. Res. 12 (4) (1964) 568581.

[24] Gillett, Billy E., and Jerry G. Johnson. *"Multi-terminal vehicle-dispatch algorithm."* Omega 4, no. 6 (1976): 711-718.

[25] Goldberg, D.E.: *"Genetic Algorithms in Search, Optimization, and Machine Learning."* Addison-Wesley, Reading (1989)

[26] Golden, Bruce L., Thomas L. Magnanti, and Hien Q. Nguyen. *"Implementing vehicle routing algorithms.* Networks 7, no. 2 (1977): 113-148.

[27] Golden, Bruce L., and Edward A. Wasil. *"OR PracticeComputerized Vehicle Routing in the Soft Drink Industry."* Operations research 35, no. 1 (1987): 6-17.

[28] Harper, Robin. *"Spatial co-evolution: quicker, fitter and less bloated."* In Proceedings of the 14th annual conference on Genetic and evolutionary computation, pp. 759-766. ACM, 2012.

[29] Ho, William, George TS Ho, Ping Ji, and Henry CW Lau. *"A hybrid genetic algorithm for the multi-depot vehicle routing problem."* Engineering Applications of Artificial Intelligence 21, no. 4 (2008): 548-557.

[30] Holland, John H. *"Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence."* (1992).

[31] Hornby, G. S., M. Fujita, S. Takamura, T. Yamamoto, and O. Hanagata. *"Autonomous evolution of gaits with the sony quadruped robot."*, In-Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, pp. 1297-1304. Morgan Kaufmann, 1999.

[32] Hornby, Gregory S. *"ALPS: the age-layered population structure for reducing the problem of premature convergence."* In Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp. 815-822. ACM, 2006.

[33] Hornby, Gregory S. *"The age-layered population structure(ALPS) evolutionary algorithm."*, (2009): 8-12.

[34] Hornby, Gregory S. *"A steady-state version of the age-layered population structure EA."* In Genetic Programming Theory and Practice VII, pp. 87-102. Springer US, 2010.

[35] Huber, Andreas, and Dieter A. Mlynski. *"An age-controlled evolutionary algorithm for optimization problems in physical layout."*, In Circuits and Systems, 1998. ISCAS'98. Proceedings of the 1998 IEEE International Symposium on, vol. 6, pp. 262-265. IEEE, 1998.

[36] Jeon, Geonwook, Herman R. Leep, and Jae Young Shim. *"A vehicle routing problem solved by using a hybrid genetic algorithm."* Computers & Industrial Engineering 53, no. 4 (2007): 680-692.

[37] Jun, Dho H. *"Multi-objective optimization for resource driven scheduling in construction projects."* PhD diss., University of Illinois at Urbana-Champaign, 2010.

[38] Karakati, Sao, and ViliPodgorelec. *"A survey of genetic algorithms for solving multi depot vehicle routing problem."* , Applied Soft Computing 27 (2015): 519-532.

[39] Laporte, Gilbert, Yves Nobert, and Serge Taillefer. *"Solving a family of multi-depot vehicle routing and location-routing problems."*, Transportation science22, no. 3 (1988): 161-172.

[40] Laporte, Gilbert. *"The vehicle routing problem: An overview of exact and approximate algorithms."*, European Journal of Operational Research 59, no. 3 (1992): 345-358.

[41] Liu, ChunYing, and Jijiang Yu. *"Multiple depots vehicle routing based on the ant colony with the genetic algorithm."* Journal of Industrial Engineering and Management 6, no. 4 (2013): 1013-1026.

[42] Naeem, Mohammad. *"Using genetic algorithms for the single allocation hub location problem."* (2010).

[43] Ombuki, Beatrice, Brian J. Ross, and Franklin Hanshar. *"Multi-objective genetic algorithms for vehicle routing problem with time windows."*, Applied Intelligence 24, no. 1 (2006): 17-30.

[44] Ombuki-Berman, Beatrice, and Franklin T. Hanshar. *"Using genetic algorithms for multi-depot vehicle routing."* In Bio-inspired algorithms for the vehicle routing problem, pp. 77-99. Springer Berlin Heidelberg, 2009.

[45] Patel, Suneer, and Christopher D. Clack. *"ALPS evaluation in financial portfolio optimisation."* In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp. 813-819. IEEE, 2007.

[46] Pisinger, David, and Stefan Ropke. *"A general heuristic for vehicle routing problems."* Computers & operations research 34, no. 8 (2007): 2403-2435.

[47] Renaud, Jacques, Gilbert Laporte, and Fayez F. Boctor. *"A tabu search heuristic for the multi-depot vehicle routing problem."*, Computers & Operations Research 23, no. 3 (1996): 229-235.

[48] Schmidt, Michael, and Hod Lipson. *"Age-fitness pareto optimization."* In-Genetic Programming Theory and Practice VIII, pp. 129-146. Springer New York, 2011.

[49] Slan, Karel. *"Comparison of CGP and age-layered CGP performance in image operator evolution."* In Genetic Programming, pp. 351-361. Springer Berlin Heidelberg, 2009.

[50] Smink, Sjoerd. *"The reality of Multi Depot Vehicle Routing models."*, Enschede: University of Twente (2010).

[51] Surekha, P., and S. Sumathi. *"Solution to multi-depot vehicle routing problem using genetic algorithms."*, World Applied Programming 1, no. 3 (2011): 118-131.

[52] Thangiah, Sam R., and Said Salhi. *"Genetic clustering: an adaptive heuristic for the multidepot vehicle routing problem."* Applied Artificial Intelligence 15, no. 4 (2001): 361-383.

[53] Van Veldhuizen, David A., and Gary B. Lamont. *"Multiobjective evolutionary algorithms: Analyzing the state-of-the-art."* Evolutionary computation 8, no. 2 (2000): 125-147.

[54] Vidal, Thibaut, Teodor Gabriel Crainic, Michel Gendreau, Nadia Lahrichi, and Walter Rei. *"A hybrid genetic algorithm for multidepot and periodic vehicle routing problems."* Operations Research 60, no. 3 (2012): 611-624.

[55] Yu, Bin, Z. Z. Yang, and J. X. Xie. *"A parallel improved ant colony optimization for multi-depot vehicle routing problem."* Journal of the Operational Research Society 62, no. 1 (2011): 183-188.

[56] Ycenur, G. Nilay, and Nihan etin Demirel. *"A new geometric shape-based genetic clustering algorithm for the multi-depot vehicle routing problem."* Expert Systems with Applications 38, no. 9 (2011): 11859-11865.

[57] http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances/

Appendix A

Additional Experimental Analysis

A.1 Empirically determining the unique parameters for the ALPS-GA

ALPS-GA differs from the traditional GA in terms of the number of layers (sub-populations) and the age attribute given to each individual in the population at a layer. The age attribute is effective after an ageing scheme and its corresponding age gap is established. We therefore experimented with three out of the four types of ageing schemes discussed in chapter 3. These are the Linear, Fibonacci and the Polynomial. We tested these schemes on the P02 instance. Table A.1 provides the results obtained after using these three schemes with an age gap of 50 and show the fitness plots for Linear and Fibonacci schemes in Figures A.1 and A.2. The fitness plot for the Polynomial is the same as that in Figure 4.4.
There was a follow up analysis on what could be the best age gap for the experiment. This was performed by using the age gaps of 20 and 60 to compare with 50 on the best established ageing scheme. This results is shown in A.2. Figures A.3 and A.4 show the graphs used for comparing the age gaps. Again these are compared to figure 4.4.

Table A.1: Comparing three ageing schemes with an age gap of 50

	Linear	Polynomial	Fibonacci
Best	476.7[5]	476.7[5]	488.26[5]
Average	516.01[5.4]	516.35[5.2]	506.13[5.27]

Table A.2: Determinimg the appropriate age gap for the polynomial ageing scheme

	Polynomial_{20}	Polynomial_{50}	Polynomial_{60}
Best	494.85[5]	476.7[5]	481.13[5]
Average	524.03[5.1]	510.12[5.2]	508.86[5.27]

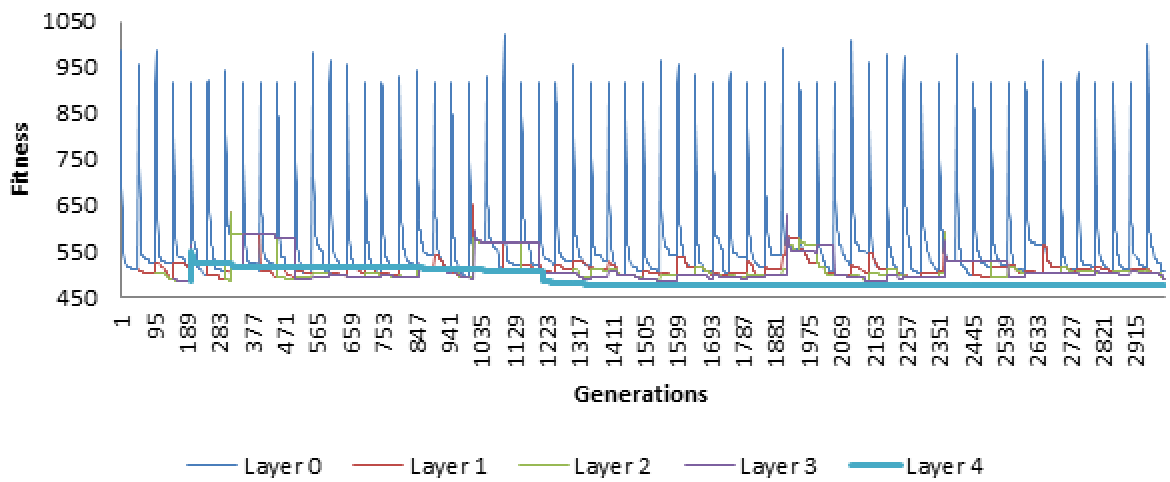

Figure A.1: Fitness plot for P02 using an age gap of 50 and a linear ageing scheme

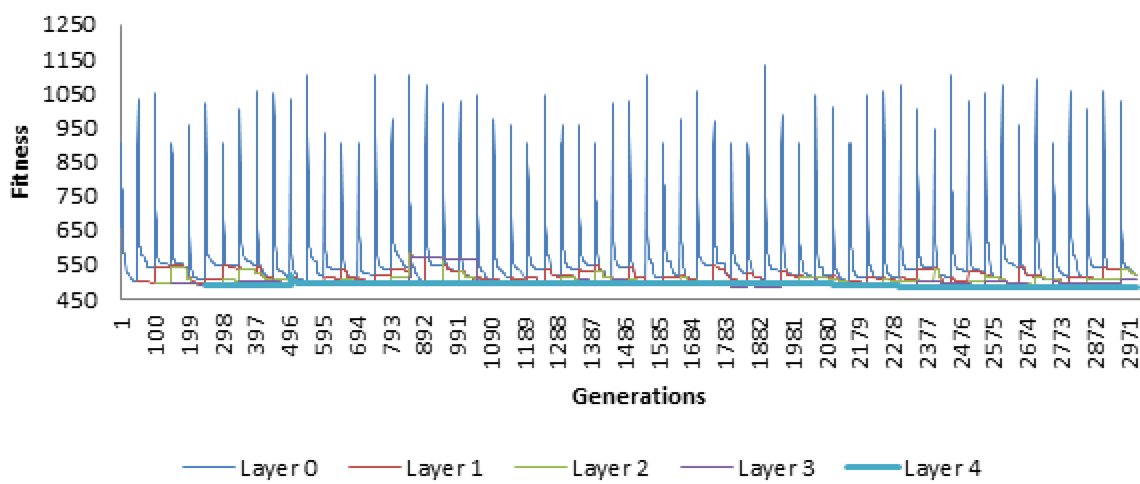

Figure A.2: Fitness plot for P02 using an age gap of 50 and a Fibonacci ageing scheme

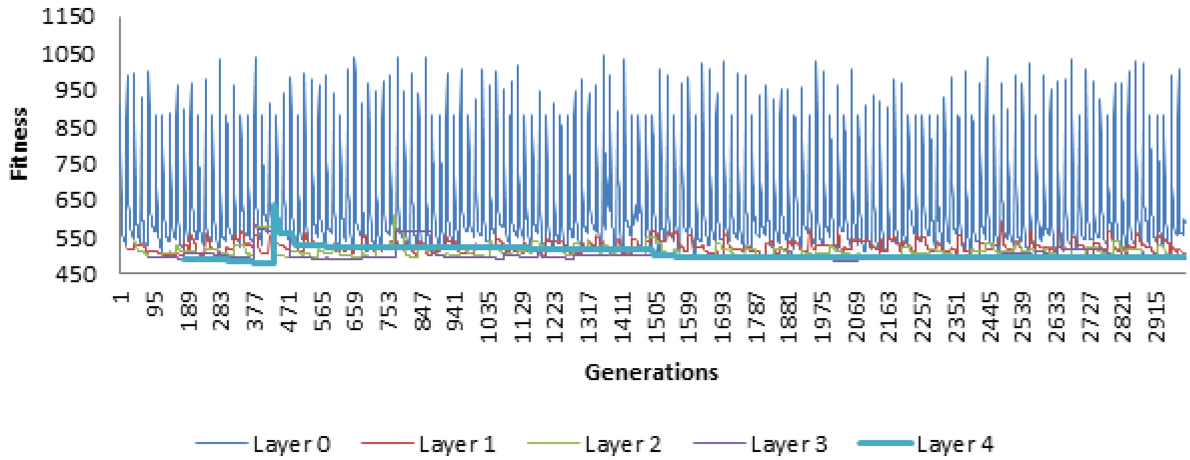

Figure A.3: Fitness plot for P02 using an age gap of 20 and a Polynomial ageing scheme

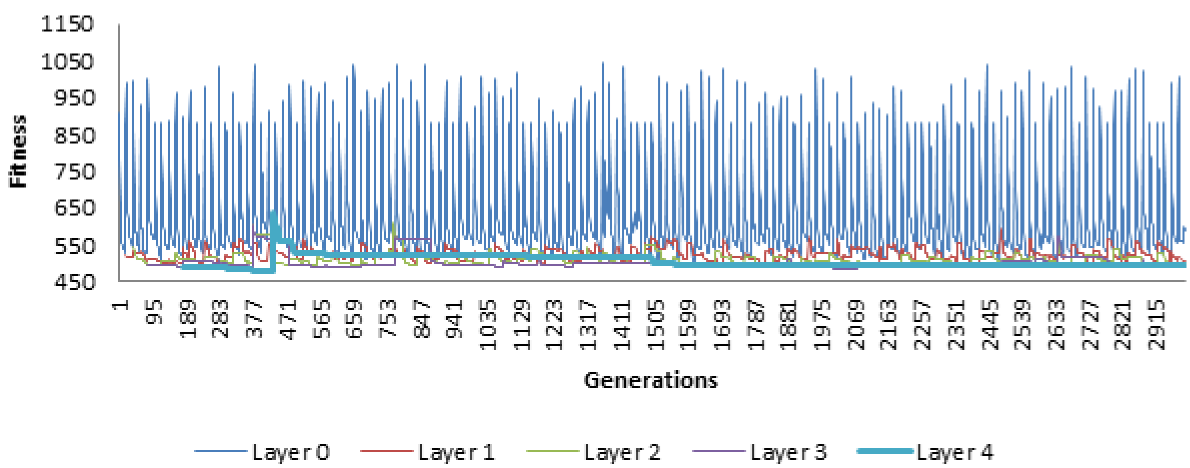

Figure A.4: Fitness plot for P02 using an age gap of 60 and a linear ageing scheme

A.1.1 Discussion: Results for ageing scheme and age gap.

From Table A.1 , it can be seen that the Fibonacci ageing scheme was outperformed by the other two schemes. First we say that these three schemes come with their independent way of setting the age limits per layer given the same age gap. Polynomial with the age limits of 50,100,200,450 for the first four layers and Linear with 50,100,150,200 win can be as a result of the individuals having ample number of generations to evolve. We did not say that it resulted from the individuals staying much longer or shorter than the other low performed scheme because of the follow up experiment we undertook. We selected the polynomial scheme based on author's preference after the analysis.
The following are the observations we drew from the follow up experiment. Most of the observations were based on the preamble that ALPS-GA allows inter layer breeding which also contribute to diversity in the system.

- If the age gap is set too high, it delays the upper layers from being active hence limiting the inter layer breeding.
- If set too low, the individuals at the lower layers have less time to evolve hence even if inter layer breeding occurs, individuals with poor fitness are considered which retards fitness improvement.
- Also if set too low, individuals at the lower layers tend to be distracted. This disruption is easily experienced in the bottom layer which is set to be reinitialized with different random seed number in a timely manner.

Appendix B

Further Analytical study

This chapter gives the tables for the significant difference analysis performed.

t-Test: Two-Sample Assuming Unequal Variances		
	Variable 1	*Variable 2*
Mean	786.0985714	774.0657143
Variance	39738.83071	37678.3155
Observations	7	7
Hypothesized Mı	0	
df	12	
t Stat	0.114419278	
P(T<=t) one-tail	0.455398797	
t Critical one-tail	1.782287556	
P(T<=t) two-tail	0.910797593	
t Critical two-tail	2.17881283	

Figure B.1: T-test for comparing non-ALPS based GA and ALPS-GA

Source of Variation	*Sum Square*	*Degree of Freedom (df)*	*Mean Square*	*F*	*P-value*	*F crit*
Between Groups	1462.9348	2	731.4673849	0.000254	0.999747	3.31583
Within Groups	86558661	30	2885288.685			
Total	86560123	32				

Figure B.2: Single factor Anova for comparing the three inter layer transfer strategies

Appendix C

Additional Experimental Analysis

This chapter show some networks as well as the tables showing averages of 30 runs for the experiments conducted.

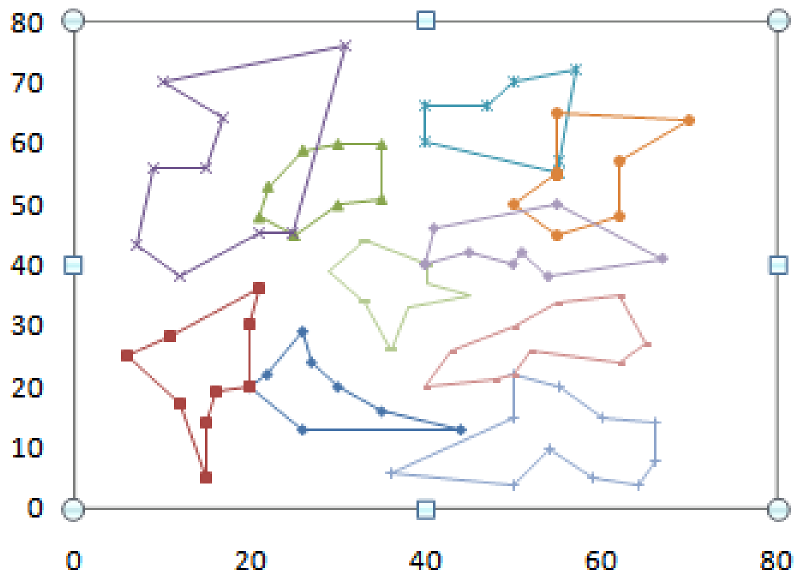

Figure C.1: Network for p03 using ALPS-GA

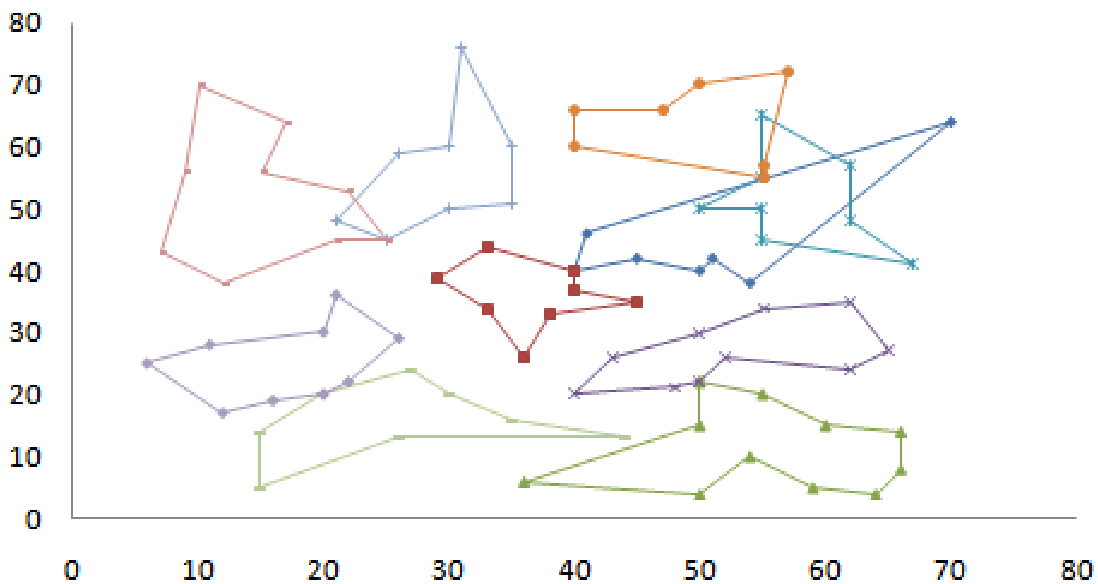

Figure C.2: Network for p03 using non-ALPS based GA

Tables C.1, and C.2 depict the averages of 30 runs per fitness evaluation strategy for the instances.
Table C.3 shows the average of 30 runs for the three inter layer transfers.
In all the tables for averages, Vehicle represents the average number of vehicles used while Distance indicate the average distance travelled.

Table C.1: Average of 30 runs for ALPS-GA using weighted sum

Instances	Vehicle.	Distance.	Instances	Vehicle.	Distance.
P01	10	618.850	P13	8	1318.95
P02	5.2	516.352	P14	8	1365.06
P03	10	717.522	P15	16	2528.71
P04	15	1076.698	P16	16	2596.84
P05	8	788.023	P17	16	2730.814
P06	15	929.242	P18	24	3903.96
P07	15.47	955.837	P19	24	3918.997
P08	25	5338.061	P20	24	4096.87
P09	25.17	4733.408	P21	36	5934.6
P10	25.27	4347.742	P22	36	5938.51
P11	25	4197.205	P23	36	6147.15
P12	8	1318.95			

Table C.2: Average of 30 runs for ALPS-GA using normalised sum of ranks

Instances	Vehicle.	Distance.	Instances	Vehicle.	Distance.
P01	10.77	601.856	P13	8	1318.95
P02	5.73	485.895	P14	8	1365.69
P03	10.9	664.982	P15	16	2615.103
P04	15	1076.93	P16	16	2618.23
P05	8	788.871	P17	16	2731.37
P06	15.37	922.976	P18	24	3935.904
P07	15.83	926.966	P19	24	3987.789
P08	25.03	5513.458	P20	24	4097.837
P09	25.83	4511.13	P21	36	5955.591
P10	25.67	4268.837	P22	36	6071.566
P11	25	4111.86	P23	36.03	6150.453
P12	8	1319.31			

Table C.3: Average of 30 runs for the three inter layer transfers using weighted sum

Instances	BIR_{veh}	BIR_{dis}	RTW_{veh}	RTW_{dis}	BTR_{veh}	BTR_{dis}
P01	10	618.850	10	619.8	10	624.66
P02	5.2	516.352	5.27	515.63	5.17	525.095
P03	10	717.522	10	720.25	10	728.38
P04	15	1076.698	15	1077.92	15	1152.502
P05	8	788.023	8	786.754	8	818.458
P06	15	929.242	15	927.30	15.03	955.90
P07	15.47	955.837	15.6	942.76	15.63	958.21
P08	25	5338.061	25	5394.490	-	-
P09	25.17	4733.408	25.2	4696.44	-	-
P10	25.27	4347.742	25.2	4314.62	-	-
P11	25	4197.205	25	4161.57	-	-
P12	8	1318.95	8	1318.95	8	1318.95
P13	8	1318.95	8	1318.95	-	-
P14	8	1365.07	8	1365.07	-	-
P15	16	2585.71	16	2587.27	16	2616.76
P16	16	2596.84	16	2604.394	-	-
P17	16	2730.814	16	2731.185	-	-
P18	24	3903.96	24	3899.823	24	3898.84
P19	24	3918.997	24	3923.224	-	-
P20	24	4096.87	24	4097.06	-	-
P21	36	5934.6	36	5921.909	36	5909.48
P22	36	5938.51	36	5945.71	-	-
P23	36	6147.15	36	6148.34	-	-

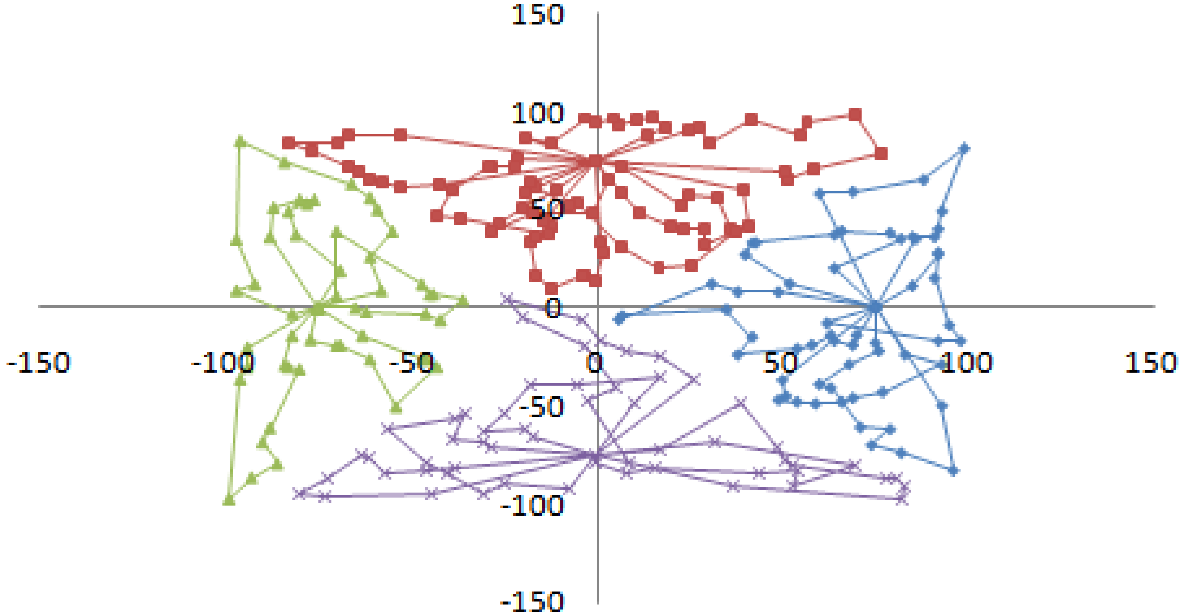

Figure C.3: Network for p10 using ALPS-GA with a distance of 4080.70 and 25 vehicles

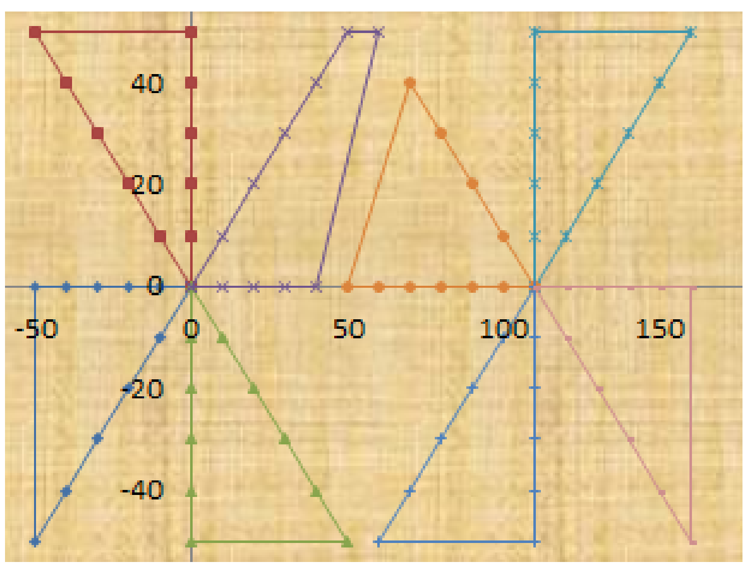

Figure C.4: Network for p14 using ALPS-GA with a distance of 1360.12 and 8 vehicles

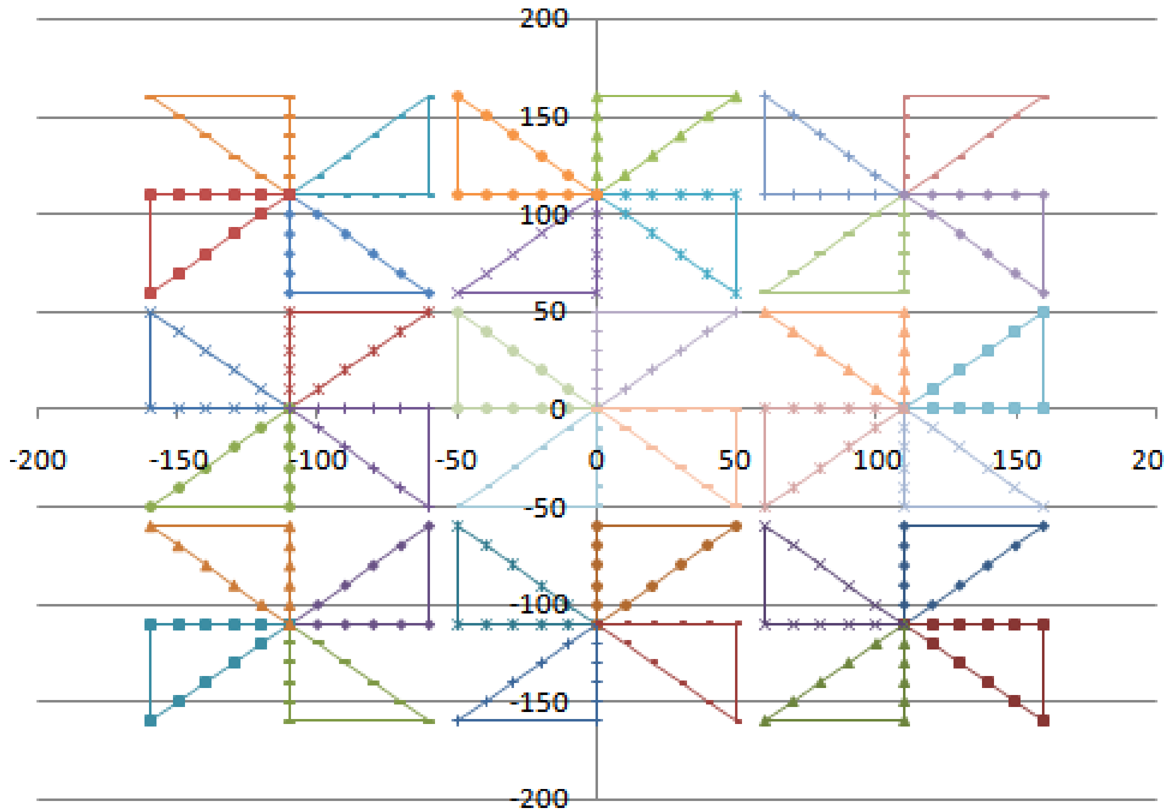

Figure C.5: Network for p23 using ALPS-GA with a distance of 6145.58 and 36 vehicles