

# LOCATION ANALYSIS OF INTERMEDIATE FACILITIES WITHIN A WASTE MANAGEMENT SYSTEM

BY

DUAN DU TOIT  
27328296

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
BACHELORS OF INDUSTRIAL ENGINEERING

IN THE

FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY

UNIVERSITY OF PRETORIA

OCTOBER 2010

## Executive summary

This report presents an introduction to municipal solid waste management and a thorough literature review on how to optimise locations of *Intermediate Facilities* in order to improve the waste management system as a whole. The importance of waste management is underlined in the introduction, together with justifications on using Operations Research to optimise waste management. The waste management system is described and the importance of *Intermediate Facilities* is highlighted. The unique scenario of the problem is explained and how the problem will be solved in colabotroration with the CSIR. In the literature review the problem is considered from the grass routes up and shown to be a Location Arc Routing Problem (LARP). The methods for solving the problem through heuristics are then discussed. A Genetic Algorithm (GA) heursitic is selected to solve the LARP and is lastly tested and taylored towards benchmark problems. The GA proved to be competent in solving the LARP, but had much randomness present in the model.

# Contents

<b>Executive summary</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background to Waste Management . . . . .	1
1.1.1 The importance of waste management . . . . .	1
1.1.2 Optimisation through Operations Research . . . . .	1
1.1.3 Waste management systems . . . . .	1
1.1.4 The importance of Intermediate Facilities . . . . .	2
1.2 Project Background . . . . .	2
1.2.1 Working in collaboration with the CSIR . . . . .	2
1.2.2 Project Aim and scope . . . . .	3
1.2.3 Research methodology . . . . .	3
1.2.4 Document Outlook . . . . .	3
<b>2 Literature Review</b>	<b>4</b>
2.1 Understanding the Problem . . . . .	4
2.1.1 Node Routing . . . . .	4
2.1.2 Arc Routing . . . . .	5
2.2 Location Analysis and Arc Routing . . . . .	6
2.2.1 Location Routing Problems . . . . .	6
2.2.2 Location Arc Routing Problems . . . . .	7
2.3 Heuristics . . . . .	7
2.4 Solution Approaches . . . . .	9
2.4.1 The Genetic Algorithm . . . . .	10
2.5 Conclusion . . . . .	10
<b>3 Model Formulation</b>	<b>11</b>
3.1 The Genetic Algorithms' Outline . . . . .	11
3.2 Generating the Initial Population . . . . .	11
3.3 Evaluating the Generation . . . . .	12
3.4 Selection . . . . .	13
3.5 Crossover . . . . .	14
3.6 Mutation . . . . .	15
3.7 Compiling the Next Generation . . . . .	15
<b>4 Computational Evaluation</b>	<b>16</b>
4.1 Computational Results . . . . .	16
4.1.1 Preliminary test . . . . .	16
4.1.2 Determining the correct population size . . . . .	17
4.1.3 Analysing a population size of 15 . . . . .	18
4.1.4 Analysing a population size of 11 . . . . .	22
<b>5 Shortcomings and Improvement Recommendations</b>	<b>26</b>
<b>Bibliography</b>	<b>28</b>



# List of Figures

2.1	Node Routing . . . . .	4
2.2	The seven bridges of Konigsberg crossing the river Pregel . . . . .	5
2.3	Arc Routing . . . . .	6
2.4	Local vs Global Optimum . . . . .	8
2.5	Total Solution Space . . . . .	9
2.6	Total Calculation Time . . . . .	9
3.1	The Procedure followed by a Genetic Algorithm . . . . .	12
3.2	Roulette Selection of IF Sets . . . . .	14
3.3	Single point Crossover . . . . .	14
3.4	Mutation . . . . .	15
4.1	GA with a population size equal to 21 . . . . .	17
4.2	Standard GA (Population size of 15) . . . . .	18
4.3	Standard GA 2 (Population size of 15) . . . . .	19
4.4	Increasing the Crossover count (Population size of 15) . . . . .	19
4.5	Increasing the Crossover count 2 (Population size of 15) . . . . .	20
4.6	Reducing the Mutation probability (Population size of 15) . . . . .	20
4.7	Reducing the Mutation probability 2 (Population size of 15) . . . . .	21
4.8	Increasing the Mutation probability (Population size of 15) . . . . .	22
4.9	Increasing the Mutation probability 2 (Population size of 15) . . . . .	22
4.10	Standard GA (Population size of 11) . . . . .	23
4.11	Increasing the Crossover count (Population size of 11) . . . . .	23
4.12	Reducing the Mutation probability (Population size of 11) . . . . .	24
4.13	Increasing the Mutation Probability (Population size of 11) . . . . .	24
4.14	Best solution convergence (GA's with a Population size of 11) . . . . .	25
5.1	Evaluating the Initial Population . . . . .	29

## Acronyms

The list of acronyms in order of appearance:

<b>OR</b>	Operations Research
<b>MWMS</b>	Municipal Waste Management System
<b>IF</b>	Intermediate Facility
<b>CSIR</b>	Council of Scientific and Industrial Research
<b>VRP</b>	Vehicle Routing Problem
<b>CPP</b>	Chinese Postman Problem
<b>RPP</b>	Rural Postman Problem
<b>ARP</b>	Arc Routing Problem
<b>CARP</b>	Capacitated Arc Routing Problem
<b>CARPIF</b>	Capacitated Arc Routing Problem with Intermediate Facilities
<b>CLARPIF</b>	Arc Routing Problem with Intermediate Facilities under Capacity and Length Restrictions
<b>LRP</b>	Location Routing Problem
<b>LAP</b>	Location Allocation Problem
<b>LARP</b>	Location Arc Routing Problem
<b>AOLRP</b>	Arc Oriented Location Routing Problem
<b>L-A-R</b>	Location Allocation Routing
<b>GA</b>	Genetic Algorithm

# Chapter 1

## Introduction

### 1.1 Background to Waste Management

#### 1.1.1 The importance of waste management

Waste management is a very important part of a municipality's service towards its residents. Besides the public's demand to have waste removed, there are two main areas that drive research attention to waste management namely: cost reduction and environmental regulations. Effective and efficient waste management will result in expenditure reductions for local municipalities. The recent recession experienced in most parts of the world caused businesses and governments to cut costs wherever possible. Waste management is in no way excluded from this tendency. The drive to cut costs together with the current trend in environmental awareness causes waste management to become an increasingly noted aspect of municipal services (Sbihi and Eglese, 2010). Extensive recycling attempts are being made, mainly by developed countries, to reduce waste and the polluting effect it has on the environment (Eiselt, 2007). The importance of proper waste management justifies the need for optimisation in waste management systems.

#### 1.1.2 Optimisation through Operations Research

Operations Research (OR) seeks to utilize science in order to derive better decisions within *operating systems* (Moder and Elmaghraby, 1978). It can be used whenever there is a problem with constraints impacting a decision. OR has its origin at the start of World War II when the British underwent a military program called *Research in Military Operations*. This is where the scientific approach of OR had its origin (Gellet, 1976). It is estimated that a savings increase of between 5% and 20% can be achieved when introducing Operations Research into transportation (Toth and Vigo, 2002). OR can therefore be used to optimise transportation systems in waste management.

#### 1.1.3 Waste management systems

A Municipal Waste Management System (MWMS) generally involves the collection of solid waste at demand points through waste collection vehicles and the transportation thereof to landfills (main dumping sites where waste is dumped at a designated site and covered with soil). Optimised MWMS have vehicles follow certain routes in a street network in order to decrease the cost of collecting waste. The collection of waste can be categorised into the following components: Commercial, Residential and Roll-on-Roll-off waste (Toth and Vigo, 2002).

- *Residential* waste is generated from households and is normally placed on pavements in plastic bags alongside houses. The requirement of the waste collection vehicle would be to service locations found along almost the entire street network in a suburb.
- *Commercial* waste is typically generated from small businesses and is usually clustered together in large waste containers near a group of businesses. If the business is large enough it

has its own waste container. The waste collection vehicle is required to serve certain points of demand, where waste needs to be collected, in a given suburb or road network.

- *Roll-on-Roll-off* waste can be generated by members of the public or commercial companies who possess large garden or industrial waste that needs to be disposed of. *Roll-on-Roll-off* waste is placed into a skip (a large container) that is mounted on top of a specially designed waste collection vehicle which transports the waste to a nearby landfill. The vehicle usually replaces a full skip with an empty skip on collection. Waste collection of this sort is done in specific areas where large amounts of waste can be gathered in one destination, therefore justifying the use of expensive *Roll-on-Roll-off* containers and vehicles.

This project will focus on residential solid waste in urban settings. Specific attention will be given to MWMSs where intermediate facilities are present.

#### 1.1.4 The importance of Intermediate Facilities

An increasing occurrence in waste management is that waste is dumped (temporarily) at *Intermediate Facilities*. Traditionally, waste collection vehicles travel to and from a landfill site, collecting waste on their way. Some landfill sites are located outside cities for environmental considerations. This results in long traveling distances for vehicles before they can start loading or unloading waste. Waste collection vehicles can alternatively dump waste at a nearby *Intermediate Facility* (IF), located in the suburb itself. Therefore the traveling cost of vehicles can be reduced by introducing IFs into a MWMS. IFs also have a large influence on which waste collection routes are optimal for vehicles to service when collecting waste in a city. If the locations of IFs in a city were to be changed, or if an IF were to be added or removed, new optimal routes would have to be calculated in order for the MWMS to remain efficient.

Another factor to be considered in MWMS is that environmental regulations have caused the number of landfilling sites in big cities to decline. Landfilling is however still a major part of waste management. According to Eiselt (2007) 55% of waste produced by American households is landfilled. Although environmental controls decrease the number of landfill sites and move them away from suburban and urban areas, the amount of waste generated by the public remains unchanged. As explained by Eiselt (2007), the decreasing amount of big landfills increases the need for IFs to compensate for the loss of landfill sites near the public. Placing IFs in suburban or urban areas are less harmful to the public health than big landfill sites. Planning the location of IFs will not only have to be done for new MWMSs but also for existing facilities, where main landfill sites are being closed due to environmental regulations. It is of great value for municipalities to be able to calculate the locations of IFs as they influence the performance and cost of the entire MWMS. This project will focus on the analysis of IFs in terms of their location and number within a given area and how that influences the cost of the overall MWMS.

## 1.2 Project Background

### 1.2.1 Working in collaboration with the CSIR

This project is done in collaboration with CSIR Built Environment. The CSIR has done extensive research on the routing of waste collection vehicles. There are numerous ways to model and optimise the routing of waste collection vehicles in a MWMSs. The CSIR's model reflects problems realistically, since they incorporate realistic constraints. These include vehicle capacity constraints, one way and two way streets, landfill site capacities and labour force shifts. Their model is able to accommodate the presence of IFs, but does not consider altering their locations to improve the MWMS. This can be done by reducing, increasing or shifting facilities around in the model and analysing its financial effect on the entire system. This project will seek to compliment the CSIR's work on the routing of waste collection vehicles by incorporating the aspect of IFs more holistically in a MWMS.

### 1.2.2 Project Aim and scope

The CSIR has build an effective optimisation model that incorporates a wide variety of realistic constraints. The CSIR's model is able to route vehicles effectively when IFs are present, however the IFs are in fixed locations and the models do not incorporate the following two aspects of IFs in a MWMS.

1. The costs involved in deploying and maintaining Intermediate Facilities.
2. How different or alternative locations of Intermediate Facilities influence the routing and overall cost of a MWMS.

The aim of this project is to account for the above mentioned aspects in an optimisation model, by analysing the number and location of Intermediate Facilities in a MWMS. All things considered the project will seek to deliver an algorithm that produces an answer to the research question:

*Where, should Intermediate Facilities be placed in order to optimise the routing and the overall cost in a given residential solid waste removal system?*

### 1.2.3 Research methodology

Since this project will make use of Operations Research techniques to derive a solution, the Operations Research Optimisation Process can be adapted in the research methodology. The process, set out by Rardin (1997) consists of four sequential steps:

1. Understand and build the model.
2. Solve the model.
3. Evaluate and test the solution.
4. Decide on implementation or re-evaluation of the model.

A similar methodology will be followed in this project.

### 1.2.4 Document Outlook

While Chapter 1 of this report gives insight to the problem at hand, Chapter 2 consists of a literature review on the problem and its solution approaches. Chapter 3 contains a formulation of the chosen model (or algorithm) to be used in solving the problem. Computational results delivered by the model is given and analysed in Chapter 4 of the document. Lastly Chapter 5 reveals a conclusion with improvement recommendations for future work.

# Chapter 2

## Literature Review

### 2.1 Understanding the Problem

The first step in the research methodology mentioned by Rardin (1997) is to understand the model or problem. Although the aim of this project is to analyse IFs in a MWMS, there should be a basis in understanding routing problems in general. The basic reason for this is that the IFs' problem forms part of the routing problem and the one influences the other.

Operation Researchers have only shown interest in modelling MWMSs in the past four decades. Generally their modelling approach involves the optimum routing of waste collection vehicles to minimise the time, distance or total cost of servicing a network of demand points (dustbins). Routing problems are generally modelled in one of two ways: Node Routing or Arc Routing.

#### 2.1.1 Node Routing

In waste collection Node Routing is typically used to model *commercial* and *roll-on-roll-off* waste collection. This is because waste is only found at certain locations within a street network. Figure 2.1 represents a typical demand network utilising the node routing method.

In node orientated routing the demand points are modelled from a graph as *vertices* (or nodes). *Vertices* also represent road crossings, junctions or dead ends in a street network. A road segment is represented by an *edge*. This formulation is typically found in the well known Vehicle Routing Problem (VRP).

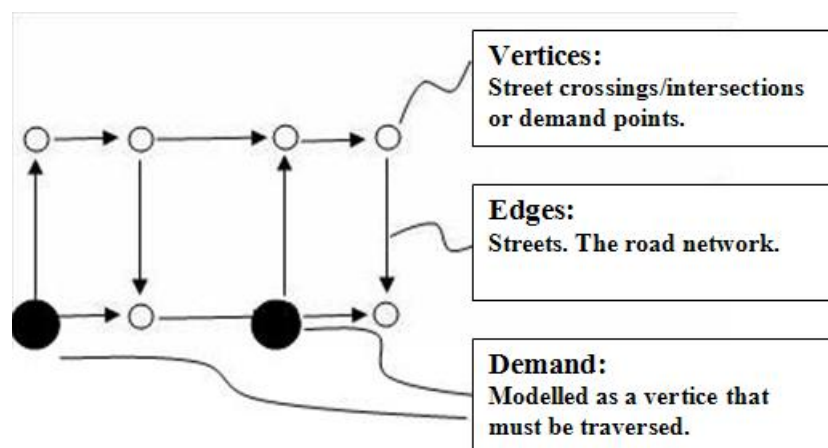


Figure 2.1: Node Routing

When solving a VRP, a set of required routes to service demand is optimised. The network's demand is serviced by a fleet of identical vehicles, where every specific route is allocated to a single

vehicle. A typical objective function of a VRP would be to minimize the total transportation cost or distance travelled. VRPs also consider operational constraints such as vehicle capacities, time windows and vehicle travelling distances (Toth and Vigo, 2002). Node Routing is preferred when modelling demand points that are far apart (e.g. pharmaceutical product deliveries to hospitals). Conversely, municipal waste in an urban setting has bins scattered all along the road, from one vertice to another. Therefore, vehicles need to service locations all along the road network (as with *residential* waste collection).

### 2.1.2 Arc Routing

The field of Arc Routing's first scientific research started two and a half centuries ago when the community of Königsberg (modern day Kaliningrad, Russia) pondered whether a couple of bridges crossing the river Pregel, could be crossed in a single walk so that the walker passes over all bridges and that each bridge is crossed no more than once. The bridges cross the river in a way depicted in Figure 2.2. The problem was given to Leonhard Euler, a Swiss mathematician who proved that it was impossible. He proved that for a closed path traversing all bridges which can be seen as vertices, each vertice must have an even degree for a closed path to exist. Such a closed path became known as an *Eulerian Path*.

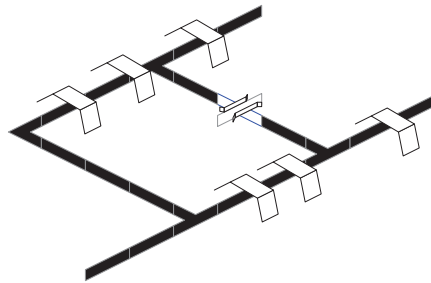


Figure 2.2: The seven bridges of Königsberg crossing the river Pregel

Euler introduced the first proven theories on arc routing. The two main classes of problems in arc routing aims at finding a route similar to an *Eulerian Path*. These are known as the Chinese Postman Problem (CPP) and the Rural Postman Problem (RPP). The CPP is concerned with finding the *shortest path* which traverses all edges *at least* once, where the RPP is similar to the CPP but only needs to traverse a subset of *required* edges. The Capacitated Arc Routing Problem (CARP) is a constraint version of the CPP or RPP (Dror, 2000).

#### The Capacitated Arc Routing Problem

The arc route modelling approach is much the same as node routing, except that an arc can span one or multiple points of demand (e.g. dustbins) between vertices. An edge set again represents road segments (Figure 2.3). A subset of arcs and edges that are *required* are then forced to be serviced by adding constraints to the model. Arcs and edges can be traversed any number of times by any number of vehicles. This is called the Arc Routing Problem in literature (ARP). When adding capacity constraints for the loads on waste collection vehicles the problem is referred to as the Capacitated Arc Routing Problem (CARP) and was first introduced by Golden and Wong (1981). The CARP has been solved with a number of approaches, both exact and approximate.

#### The Capacitated Arc Routing Problem with Intermediate Facilities

When including *known* IFs as a subset of locations, the CARP is altered to the Capacitated Arc Routing Problem with Intermediate Facilities (CARPIF). This was introduced by Ghiani et al. (2001). The location of IFs is known and fixed in the formulation of the problem. The CARPIF allows vehicles to offload waste at an IF, which is treated as a normal dumping site in the model. The vehicles can do so provided that the accumulated demand carried by the vehicle before

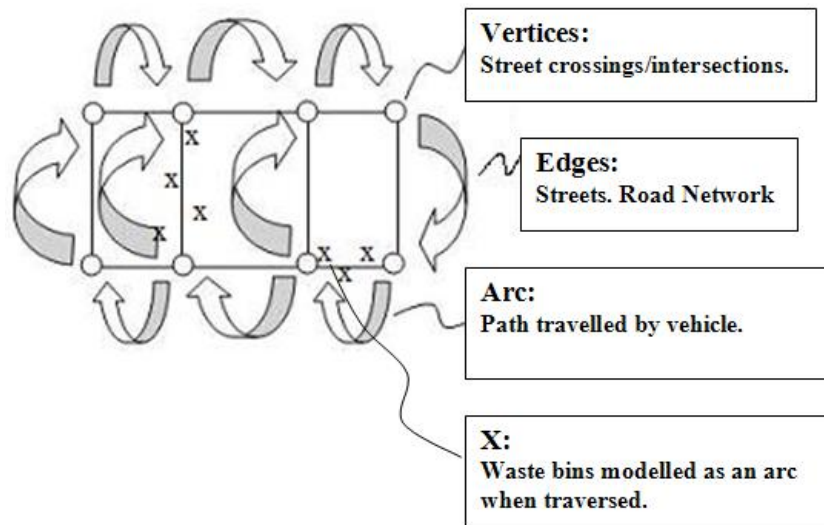


Figure 2.3: Arc Routing Model Approach

dropping waste at an IF or landfill site does not exceed the capacity constraints of the vehicles. Ghiani et al. (2004) later added length restrictions to the CARPIF and derived the name: *Arc Routing Problem with Intermediate Facilities under Capacity and Length Restrictions (CLARPIF)* which is a more realistic formulation of the problem. The length restrictions restrict the tour (or traversal) length a waste collection vehicle can undertake and is therefore an improved model of the CARPIF (Ghiani et al., 2001). The Arc Routing Problem and the Vehicle Routing Problem are related to each other and each can be transformed into a similar problem of the other class.

### Transformations of ARP to VRP

The field of arc routing is gaining momentum. However, since more literature is available on Node Routing Problems, some have ventured to turn Arc Routing Problems into Vehicle Routing Problems. The first and most generally used method for converting ARPs into VRPs was done by Pearn et al. (1987). In the model each required CARP *arc* is transformed into three Capacitated VRP nodes. The transformation entails certain drawbacks (Letchford and Oukil, 2009):

1. A huge amount of memory is needed to perform the transformation.
2. The transformation method is not suitable for all types of graphs.

Given the complexity and size of a realistic MWMS, it was decided to align with the CSIR, in their Arc Routing modelling approach.

## 2.2 Location Analysis and Arc Routing

Determining the location of facilities in waste collection type problems is done in one of two ways according to the routing orientation followed (node or arc routing) in the model. The two approaches are known as Location Routing and Location Arc Routing Problems.

### 2.2.1 Location Routing Problems

The Location Routing Problem (LRP) is a combinatorial problem, optimising facility locations and vehicle routes at the same time. LRPs follow a node routing formulation and is described by Liu et al. (2009) as the combined Location Allocation Problem (LAP) (see Cooper (1964), Cooper (1963) and Miehle (1958)) and the Vehicle Routing Problem (VRP) (see Dantzig and Ramser (1959) and Gendreau et al. (1994)). The LAP determines: the number and location of facilities,

the allocation of demand to facilities and lastly the quantity to be transferred to facilities. The LAP does not utilise the arc oriented modelling approach, but can be adapted to integrate with arc routing in an urban MWMS. The routing part of a LRP is a Node Routing Problem, where nodes represent customers to be serviced. In Arc Routing customers are represented by an arc (which is a better formulation of the problem in residential waste collection) (Levy and Bodin (1989)).

### 2.2.2 Location Arc Routing Problems

The *Location Arc Routing Problem* (LARP) is similar to the LRP, but is modelled in an arc oriented way. Ghiani and Laporte (2001) state that the LARP is encountered when it is needed to simultaneously minimise a tour of required edges and arcs in a network *and* to place facilities according to a range of constraints. Like the ARP, the LARP can be used in waste collection or in snow clearance and postal deliveries (Ghiani and Laporte (2001)). The LARP can be seen as an integration of the *Capacitated Arc Routing Problem* and the *Location Allocation Problem*. As noted by Ghiani and Laporte (1999) and Liu et al. (2009), this is a small field of study with a small amount of literature available. This scarcity in literature can be attributed to the difficulty in solving the LARP, because both the CARP and the LAP are in themselves difficult to solve.

The LARP was introduced by Levy and Bodin (1989) after involvement in a postman scheduling problem. In the problem the postman parks his van at a street intersection and then delivers mail by walking an Euler path. The problem was coined the Arc Oriented Location Routing Problem (AOLRP). It was only later that the term LARP became universal. The AOLRP can be viewed as a LARP for it consists of a location problem and a routing problem simultaneously. The intersection where a postman is able to park his vehicle can be viewed as the location of a facility and the Euler path traversed represents an arc routing problem (Levy and Bodin (1989)).

The AOLRP was solved in three steps which forms the basis of the LARP's solution methodology:

1. Determining facility location
2. Allocation of arcs into partitions
3. Route optimising

Ghiani and Laporte (2001) who surveyed the major applications of the LARP mention that when the locations of IFs (or depots) in waste collection is known and fixed the LARP filters to a normal CARP and can be solved as a CARPIF or CLARPIF. It is only when the locations of facilities need to be determined that the problem becomes a LARP. The solution approaches used in solving the LARP revolve around heuristics which breaks the problem into the three aspects of its inherent nature (as done with the AOLRP by Levy and Bodin (1989)). These three aspects are: Location, Allocation and Routing (L-A-R). The order in which the aspects are done may be swapped around (for example a Routing Allocation Location (R-A-L) formulation can be followed). In this project the Routing and Allocation of Routes to customers aspect has been done by the CSIR. Their model is able to calculate optimised routes as soon as it knows where facilities are located. Therefore the methodology to be followed in this project will be L-A-R. To solve the LARP through a L-A-R methodology, further heuristic methods are used within the location and routing aspects themselves. The routing aspect done by the CSIR encompasses the *allocation* aspect into the routing and can therefore be seen as one aspect called routing.

## 2.3 Heuristics

Heuristics are used to find approximate solutions when optimal solutions can not be found. Approximate solutions are used typically because of time constraints when solving complex problems. Figure 2.4 illustrates the difference between local optimum solutions and global optimum solutions. Both the CARP and the LARP (being an extension of the CARP) are too complex to solve with normal Linear Programming formulations. This is because the CARP has an exponential order of computations. Therefore, there exists no known polynomial algorithm that can solve it

to optimality in a reasonable time (Winston and Venkataramanan, 2003). The CARP is therefore termed *NP*-Hard (non deterministic polynomial-time hard). Golden and Wong (1981) showed that even finding a 0.5-approximation of the global optimum is *NP*-Hard. By means of relaxation, exact solutions can be found. Although relaxation can be used to determine global optimum solutions, the method is only fit for small problems as noted by Liu et al. (2009). Therefore heuristic or approximate solutions will give better answers to real life problems.

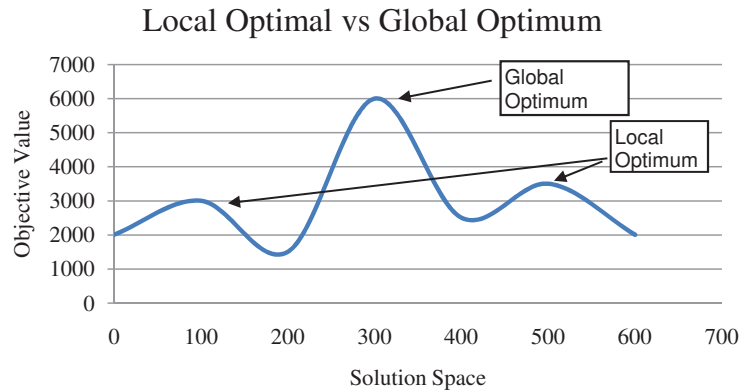


Figure 2.4: Local vs Global Optimum

The difficult part of solving the LARP is found in the routing aspects of the problem. There are a range of heuristic and meta-heuristic methods used in the literature including: Variable Neighborhood (Hansen et al. (2001)), Tabu Search (Hertz et al. (2000), Genetic Algorithm (Prins et al. (2003))). These extensions allow searches to accept iterations that result in nonimproving objective values. Iterations of this sort are called *nonimproving moves* which permits the algorithm to pass through local optimums towards global optimums. Allowing *nonimproving moves* may cause the algorithm to search in the same solution space (ie. get stuck in repeating moves). Most heuristics have a way of countering this problem. The CSIR have based their routing model on the Tabu Search heuristic. The Tabu Search prohibits the algorithm from repeating the same searches in the solution space by declaring certain moves temporarily tabu. Determining the locations for the routing model is not as complex and it is therefore not necessarily needed to develop such complex heuristics.

Levy and Bodin (1989) made use of three criteria to place locations in a preference list. This list annulled some possible locations that did not conform to the criteria. The criteria used to place intermediate facilities are:

1. A facility can be too close to a depot (in terms of the number of arcs between the intermediate facility and the depot).
2. The Intermediate Facility can not be placed too close to a border.
3. Each possible Intermediate Facility is given an attractiveness rating of some sort (The rating is for the demand met by the facility, but is problem specific).

The facilities that met the criteria were then placed according to their ratings on a hierarchical list. Choosing one location from the list at a time, the routing aspects were calculated and compared with each other. IFs were then placed on a short list of four IFs, from where the location producing the best objective value was chosen. The problem analysed by Levy and Bodin (1989) ended up accepting only one IF into the problem. This project however will look at more locations and possibly end up with more than one accepted intermediate facility.

## 2.4 Solution Approaches

Approximate solution approaches will have to be followed because of time constraints in calculating the LARP. Following the L-A-R methodology, it must be noted that routing algorithms run within a range of two to twenty minutes depending on the desired quality of the solution. When considering a few possible IF locations the problem can simply be solved by calculating each possible solution independently and then choosing the best one. The number of possible solutions when considering two IF locations is 4 (approximated by the formula  $2^x$ , with  $x$  the number of possible IF locations). An IF has two possible states in the model, it can either be chosen or it can not be chosen at a specific IF location. The total time that it would take to evaluate all possible solutions for two IFs is therefore  $2 \text{ min} \times (2 \text{ possible states to the power of } 2 \text{ IF locations considered}) = 8 \text{ min}$ . The four evaluated IF solutions each produce an objective function value, of which the maximum reveals the global optimum solution for the LARP. This can however not be done for bigger problems, because of time constraints. Figure 2.5 indicates the exponential growth of possible solutions with increasing possible IF locations. This translates into a huge amount of time needed to analyse all possible solutions as shown in Figure 2.6.

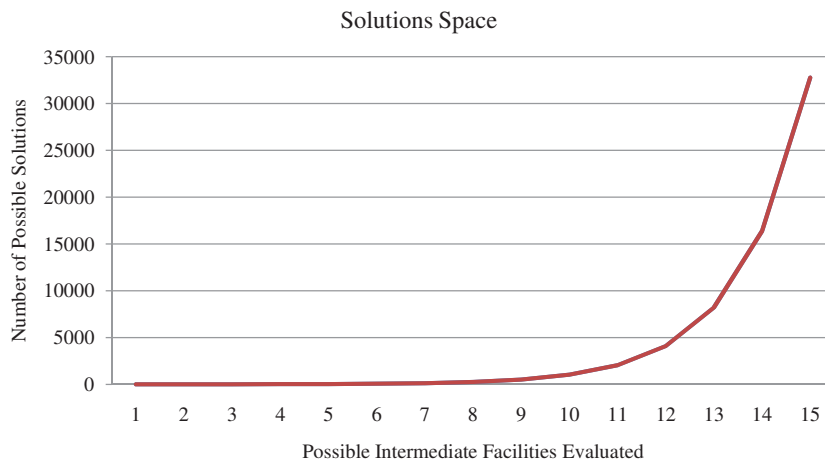


Figure 2.5: Total Solution Space

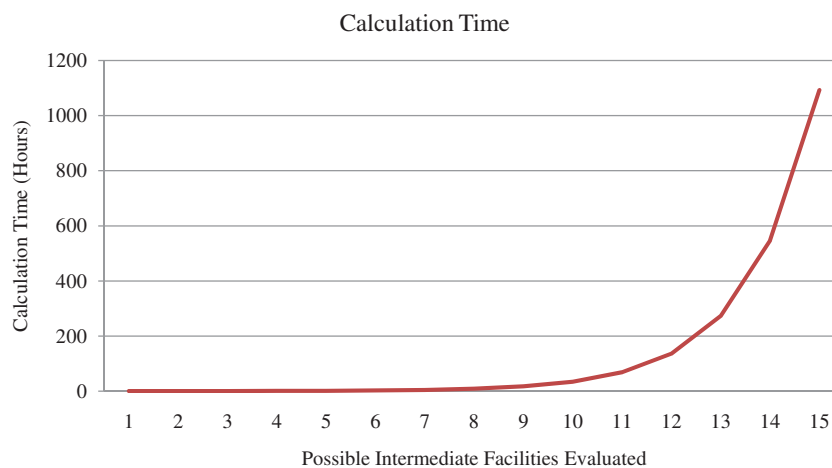


Figure 2.6: Total Calculation Time

Therefore this problem is too large to be solved through numerical analysis, whereby all solutions are observed. Also the risk of ending up with a local optimal solution, when not utilizing

nonimproving search techniques, calls for more advanced heuristics.

### 2.4.1 The Genetic Algorithm

The Genetic Algorithm (GA) is a widely used heuristic in literature. It seeks to copy the concept of biological evolution to converge toward a global optimum. GAs have been applied to many problems in a wide range of study fields including: engineering, social studies, bussiness, and mathematics (Pang, 2006). Certain problems have been robotics, aeronotics, jobshop scheduling and facial recognition (Coley, 1999). A GA is relatively easy to program and it fits the problem at hand because it easily utilizes binary numbers, like open or closed IFs would be in the model.

The GA starts with a *population* of possible solutions, termed a *generation* (which is the same as a gene pool in biology). Each solution is called a string in the algorithm (which is the equivalent of a chromosome in biological terms). It then calculates the objective function of each solution and selects the next generation according to an evolution like criteria. The best solutions are transferred to the next generation (the population's *children*), after being varied by the heuristic. The variation procedure combines some of the current generation's solutions with each other, aiming to produce improved solutions into the child's population. The remainder of the children's population will then consist of generating new solutions randomly or it will follow a case specific heuristic. Repeating this procedure converges solutions in *generations* towards a global optima. Because of the randomness involved in mixing solutions, the possibility of converging towards a local optima or only searching through secluded parts of the solution space is significantly reduced.

## 2.5 Conclusion

In the light of the literature reviewed the following can be concluded. In developing an optimised waste management system by focussing on *Intermediate Facilities*, the routing of waste collection vehicles must be taken into account, as the one influences the other. Therefore the problem is a LARP and can be solved by breaking up the problem into it's three subcomponents (location, allocation and routing). Since the CSIR's model provides the routing and allocation aspect of the LARP, IF locations can be manipulated in a custom made separate model, then be fed back to the routing model. The routing model will then recalculate the routes and allocate customers to routes, thereby giving a new objective function. In other words the project will follow a L-A-R methodology. The location aspect of the LARP was shown to be complex and time consuming. Therefore heuristic methods must be used. The heuristic method used in this project is the Genetic Algorithm. The effectiveness of different variation criteria and constraints will be tested against each other with regards to computation time, proximity to a global optimum solution and the ability of the algorithm to search through the entire solution space.

# Chapter 3

## Model Formulation

### 3.1 The Genetic Algorithms' Outline

The model formulation approach followed in this project is similar to most GAs and can be simplified into six steps. The six steps in the GA are to: generate an initial random parent population (the first generation), evaluate the generation, select strings from the parent generation for variation, cross-over the selected strings, mutate the strings and lastly, insert the strings into a new child generation.

These steps have been depicted in Figure 3.1 below as they occur in the model with steps 2 to 6 forming a big evolving loop. New cycles occur in the loop when a child generation goes through the evaluation step as a newly created generation. The big loop will continue until a certain amount of generations have passed. After step 2 in the model, the generation enters the variation procedure (or small loop). The variation procedure starts at step 3 which selects only two strings (chromosomes) out of the generation at a time. The selected strings then go through steps 4 and or 5. This is because a crossover limit will prohibit strings from undergoing crossover after being reached and therefore strings will pass on directly to Mutation. When strings have been varied they are transferred into a child generation. This variation procedure will continue until the child generation is completed with pairs of adapted strings.

As shown by the literature study, GA's can be used in a diverse range of problems. Once a unique problem is identified, the general GA can be adapted to improve the algorithms' effectiveness in solving that specific problem. The more tailored an algorithm is towards a specific problem, the better its effectiveness in finding a quality solution. The extent to which this can be done lies mainly in the fine tuning of steps 3 to 5 (the variation procedure) in the algorithm. We will now look more closely at how each of the six steps in the model formulation have been programmed.

### 3.2 Generating the Initial Population

The first step in the model is to generate a completely random population of solutions in the form of binary strings. Strings consists of binary bits and each bit represents either an open or a closed IF site (1 or 0 respectively). The length of every string is equal to the number of possible IF locations considered in the problem. For instance if 4 possible IF locations are considered in a residential area, then each string in the population would consist of 4 binary numbers. The model is programmed to automatically detect the number of IFs considered in a given problem by analysing the problem's input data (obtained from the CSIR). The population size is made to be a variable that can be specified for sensitivity analysis purposes in the model.

When the first generation of solutions have been randomly generated, the algorithm will then enter into a big evolving loop. This loop consists of continual evaluation of varied generations.

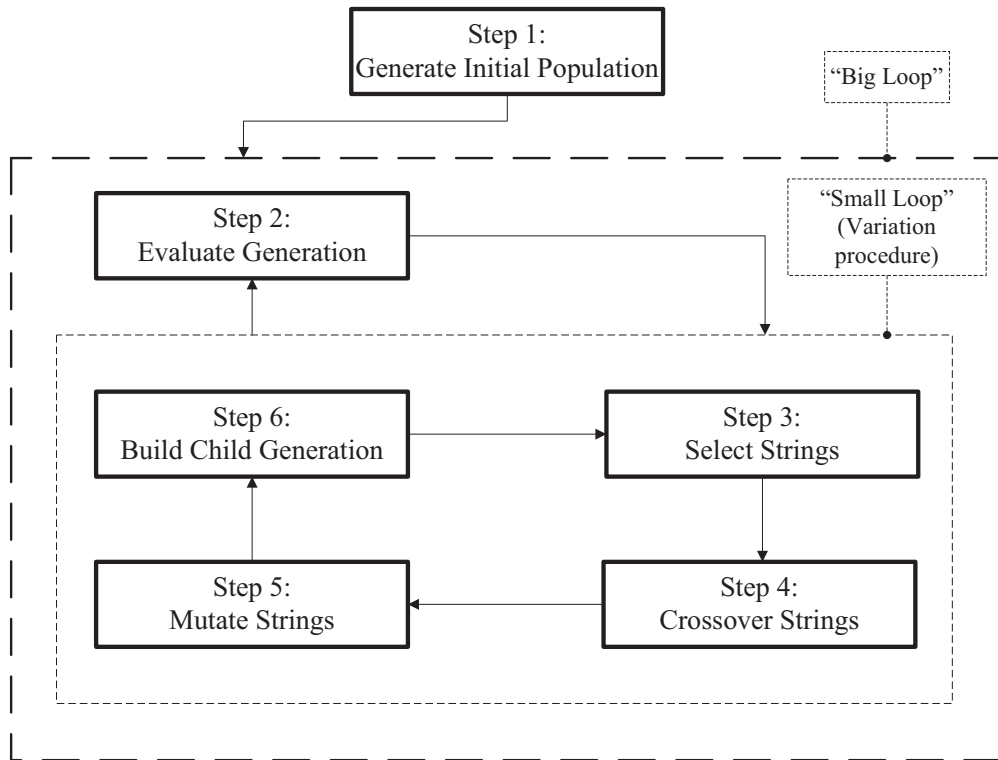


Figure 3.1: The Procedure followed by a Genetic Algorithm

### 3.3 Evaluating the Generation

Each new generation is evaluated by what is called the fitness function. The fitness function is actually the same as the objective function in the model and measures the following three aspects:

1. The routing cost of vehicles in the MWMS, given a set of IF locations. This includes the traversal cost of required and non-required arcs and edges by vehicles, as well as the costs involved when unloading waste.
2. The fixed cost of erecting additional IFs in the MWMS.
3. The variable cost of all the IFs in the proposed solution.

The routing cost can be seen as the routing and allocation aspect of the LARP's L-A-R methodology (Chapter 2.2.2) and is done by calling the CSIR's model in the program. This is done for each string in the generation as the routing model only considers one string (or IF set) at a time. Only open IF locations (equal to 1 in the string) are evaluated and the remaining bits (equal to 0) are ignored when calling the routing model. For example a possible solution might be the string [1 0 0 1 1 0 0 0 1] when looking at ten possible IF locations. The algorithm then filters through the string to determine which IF numbers are open in the string. In this case these are [1, 4, 5 and 10]. This shortened string is then fed into the routing model which knows where the IFs are located in the street network, in this case it might be on nodes [7, 31, 80 and 174] as an example. The routing model then returns the total routing cost for that specific solution in the generation. As was shown in the literature study, this part of the problem is actually a CARPIF (Chapter 2.2.2). Here fixed intermediate facilities are represented by the proposed IF set.

Additionally if new IFs are present in the proposed solution then erecting costs are added to the specific string evaluated after its routing cost has been established. Each new facility is taken to cost R1 million. Spread out over a five year period, assuming that there are 200 working

days in a year and that the minimum accepted rate of return is 12 percent, amounts to an annuity cost  $P(A/P, 12\%, 5)$  of R1387 per day. The salvage value of each open IF is also taken into account since an IF can be seen as an asset. According to the ABSA house price index it was derived that an annual growth of 5% can be expected for IFs after inflation. This gives an annuity income  $P(A/P, 5\%, 5)$  of R1155 per day. Subtracting the investment cost of opening up a new IF from its salvage value gives a net fixed cost of R232 per day per IF. Existing facilities would be favoured by the model since they have no fixed costs to be added to the function.

The variable costs of facilities constitutes of staff that help unload the residential waste gathered by waste collection vehicles and some management who would oversee the IF. This cost is estimated to be R600 per day. Since labour cost rises with inflation an annuity has to be calculated based on an increasing geometric gradient series caused by the rising labour cost. Labour cost is taken to rise with the same amount as the minimum accepted rate of return ( $i=g=12\%$ ). The present worth of all the increasing labour cost can be calculated from  $A(n/(1+i))$ , with  $A=R600$ ,  $n=5$  and  $i=12\%$ . This amounts to R2679. Therefore an annuity  $P(A/P, 12\%, 5)$  of R743 was calculated based on the present worth of the increasing geometric series.

Altogether the fixed cost of R232 for each new IF, the variable cost of R743 for each open IF and lastly the routing cost of the given IF set constitutes the net fitness function of a string. The net fitness function of each string in the population is then saved in the model. The average as well as the best and worst fitnesses in the population are then derived and stored for future use. Minimising the fitness function by changing IF locations is the objective of the model. Changing IF is done by the variation procedure which will be explained in the following four sections.

### 3.4 Selection

Once the population's fitness is calculated strings are then selected in pairs according to an evolution-like criteria. Pairs of strings are selected, varied and transferred into the population's Child generation. This procedure will repeat itself until the child population has reached the same number as the parent population.

There are many methods used for selection but this project settled on the more popular Roulette Selection Method. Roulette Selection increases the probability of better solutions to be selected for variation and transfer into the child. It is based on the gambling game *Roulette*, where a rotating disc with slots is spun, bets are then made on which slot a small ball will rest in. The roulette method is graphed in Figure 3.2 below and shows how solutions with high fitness values have bigger slots and therefore have a higher probability to be chosen for variation. This is accomplished in the model through repeating the following procedure:

- Divide each string's fitness function by the sum of the generation's total fitness.
- Save these fractions of the fitness function as vector components into a new vector. The vector components represent the slots of the roulette disk, with higher fractions occupying bigger slots of the disk.
- Generate a random number between 0 and 1 that will decide which string must be selected. The random number is the equivalent of the disk spinning and the small ball resting in a slot.
- Select the vector component in which interval (or slot) the random number lies in.

By going through this procedure two times, the model selects the pair of solutions from the generation that will undergo variation. Apart from Roulette Selection, the best solution is then also transferred to the child generation. This is called *Elitism* and is done by copying the minimum fitness function's string from the parent generation into the child generation. This is done once for every generation in the big loop. Elitism is true to Darwin's survival of the fittest theory which is exactly what GAs aims to mimic. Thus, by transferring the minimum objective value of the parent population into the child population, the best solution is always kept.

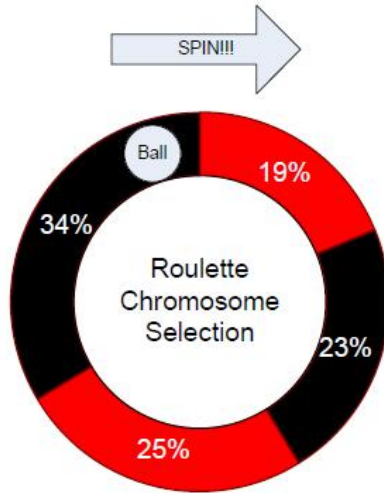


Figure 3.2: Roulette Selection of IF Sets

### 3.5 Crossover

Too much variation can diminish the convergency of the heuristic. Variation however, is needed and serves as a counter for convergence towards local optimas in GAs. Therefore a fine balance of variation and convergence is desired. Crossover is the first of two methods used to increase variation in the population. Roulette selected pairs of strings cross over certain parts of each string with each other. This model made use of the common single point Crossover method, where a randomly selected point (somewhere in the range of the string length) determines that every binary bit after that point must be crossed over to the other strings binary bit(s) in the same position. Figure 3.3 below serves the purpose of showing the Crossover technique. In the figure two strings (with length equal to 10) have been selected from the parent population to Crossover and can be seen as the original pair. The Crossover point is between bits 6 and 7, which means that the last two bits of each string need to be swapped with each other. Creating two new strings ready for Mutation. The number of strings that undergo Crossover can be changed in the model for sensitivity alaysis. Strings that do not undergo Crossover are directly taken to the Mutation procedure when the Crossover limit has been reached. All strings that undergo Crossover in the model also have to go through Mutation.

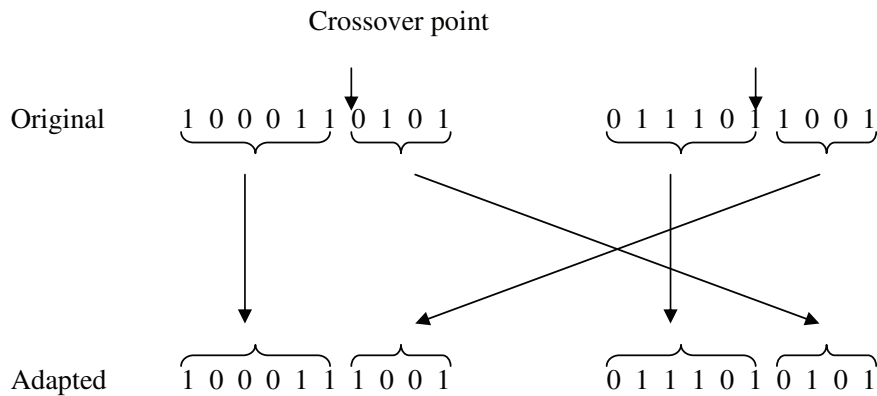


Figure 3.3: Single point Crossover



# Chapter 4

## Computational Evaluation

### 4.1 Computational Results

In this chapter the model is evaluated in terms of its effectiveness in solving the LARP. An initial test was conducted to establish whether the created GA could solve a LARP, using the *lpr-a-04.dat* benchmark problem with added IFs. After the GA proved its capability, more extensive analysis was executed. It was assumed that the population size of the GA had the greatest influence on the model's effectiveness. Therefore, two different population sizes were used to conduct a sensitivity analysis on the GA proposed. These were a population size of fifteen and ten. Other variables that influence the GA's effectiveness (the Crossover Count and the Mutation Probability) were also tested at each of the two population sizes. After the first population was tested it was found that there is extensive randomness present in the GA. This was reduced in the second population size of ten's tests by prescribing the same initial population to each of the GA variants in the sensitivity analysis. Thereby removing the effect that starting with a good or bad population would have on the GA's future generations.

#### 4.1.1 Preliminary test

##### **Adapting the benchmark problem and the criteria used to evaluate the GA**

The *lpr-a-04.dat* benchmark problem has existing IFs as part of the street network on the following nodes: [97, 194]. A total of 8 possible IF locations were added in order to enlarge the problem, reflecting a typical decision faced by a MWMS. The problem therefore consisted of existing and possible IF sites. The locations of the IF contained in the street network were respectively placed on nodes:

- Additional (possible) IF locations: [13, 34, 55, 76, 98, 99, 110,160]
- Existing IF locations: [97, 194]

The additional IFs were randomly selected. Real world problems will differ from each other with regards to where IFs are able to be erected and will depend on the geographical area considered. The criteria used in determining the effectiveness of a GA however, will stay relatively the same. The following criteria were used to evaluate the GA:

- Solution Cost:  
Is it a quality, low cost solution to the LARP? Considering that an optimal solution might not be achieved, the GA should provide a relatively good solution *within time constraints*.
- Convergancy:  
Does the algorithm evolve solutions over time so that they converge towards an optimal solution? This is measured in terms of the population's best.

- Unstable variation:

In terms of the population's averages (excessive variation leads to unstable variation). Does the population's average fitness converge as a whole?

- Stable variation:

In terms of the population's worst solutions. Is there enough variation to cover the entire solution space?

#### 4.1.2 Determining the correct population size

As mentioned in Chapter 3, the extent to which a GA is tailored lies mainly in the following variables: Population size, Crossover count and Mutation probability. Preliminary tests were done to establish the best Population size first with specific attention given to the time the GA takes to arrive at a good solution. A notation was used to easily distinguish between variants of the GA proposed. A GA that is computed with 39 iterations with a population size of 21, a Crossover count of 2 and a Mutation probability of 0.1 will have a notation of GA (39/21/2/0.1). The iteration number refers to the amount of child generations created in the big loop of the algorithm (Chapter 3.1).

Figure 4.1 shows the computational results of a GA with the following variables:

- Population size = 21
- Number of strings through Crossover (Crossover count) = 2
- Mutate probability = 0.05

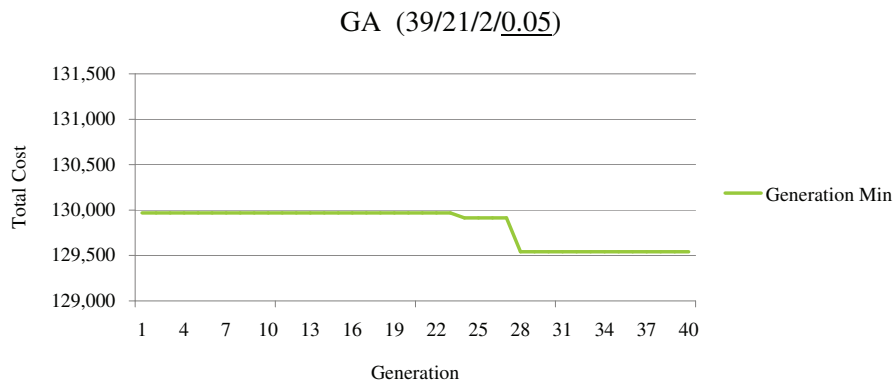


Figure 4.1: GA a with population size equal to 21

The GA used to obtain the results shown in Figure 4.1 ran for a total computational time of 28 hours. At this point it is important to notice that the GA arrives at a minimum total solution cost of R129542 after 28 iterations. One iteration of the GA consists of calculating the routing cost of every string in a population of 21 strings. Therefore one iteration of a GA with a population equal to 15 will be much shorter than calculating one iteration of a GA with a population size of 21. This preliminary run shows that the GA considered took 15 hours to arrive at this solution. The solution cost of R129542 will further on in the document be shown to be a good solution, since most variants of the GA created reaches solutions close to this. This already shows that the GA created is faster in getting a quality solution than enumeration methods would have taken to arrive at an optimal solution. Enumeration would have taken 34 hours to calculate every possible solution for a problem with 10 IF (see Figure 2.6 and Chapter 2.4). This is 19 hours longer than the 15 hours taken by this GA to reach the solution cost of R129542. The effect of using smaller population sizes proved to further enhance the GA and will be discussed in the next two subsections of the computational results acquired.

### 4.1.3 Analysing a population size of 15

The initial results showed that the GA successfully solved the benchmark data and that the *Elite* string (or the generation's minimum) does converge over generations. More extensive sensitivity analysis was done on GAs with a population size of 15. Four different GAs were tested against the benchmark data on two occasions.

The first GA was used as a median from where a sensitivity analysis was performed as it consists of standard accepted variables. Figure 4.2 graphs the results of the first GA. The GA was computed in 39 iterations with a population size of 15, a Crossover count of 2 and a Mutation probability of 0.1 hence the notation on the graph: GA 1 (39/15/2/0.1). The other three GAs were adapted in terms of an increased Crossover count (Figure 4.4), a reduced Mutation probability (Figure 4.6) and an increased Mutation probability Figure (4.8).

#### Standard GA

From Figure 4.2 we observe that the first result showed a very quick convergence of the generation's minimum objective value. This seems to be unusual when compared to the other results obtained and might be because of the intrinsic randomness of the GA that it arrived at such a good solution in such a short period of time.

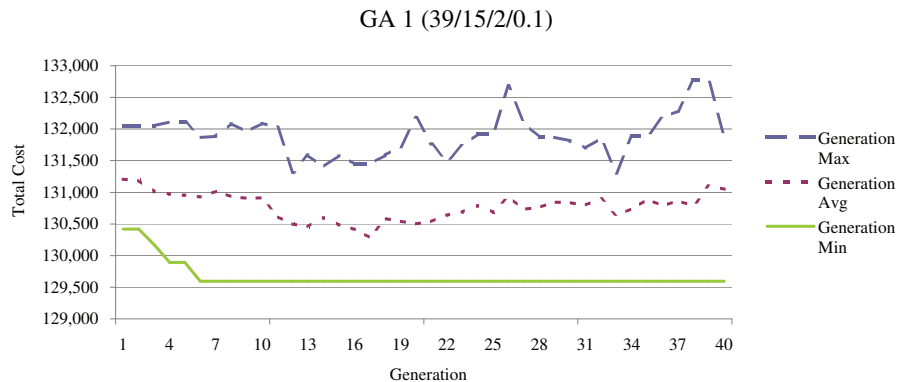


Figure 4.2: Standard GA (Population size of 15)

- Minimum Objective Value = 129,594
- IF set proposed = [1.0.1.0.1.0.0.0.1.0.]
- Computation time to arrive at this solution (h) = 3.15
- Convergancy: Unusually fast
- Unstable variation: The population's average converges slightly over generations.
- Stable variation: There is an abundance of variation in the generation's worst objective functions.
  
- Minimum Objective Value = 129,542
- IF set proposed = [1.0.0.1.0.0.0.0.0.0.]
- Computation time to arrive at this solution (h) = 9.975
- Convergancy: Good
- Unstable variation: The population's average does converge effectively over generations.
- Stable variation: There is an abundance of variation in the generation's worst objective functions.

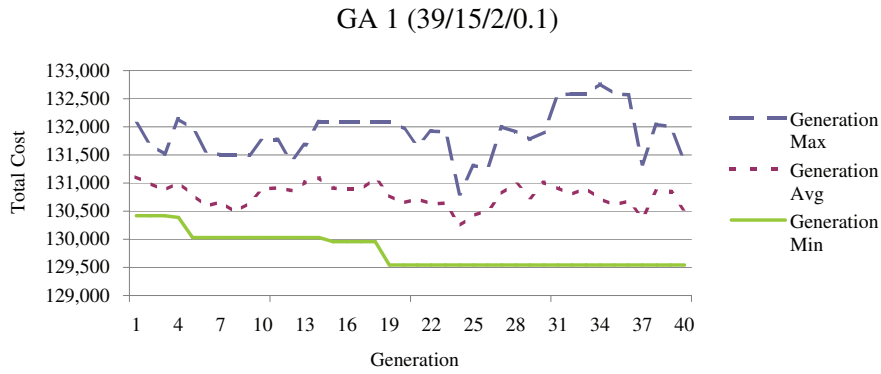


Figure 4.3: Standard GA 2 (Population size of 15)

### Increased Crossover count

From Figures 4.4 and 4.5 we can observe that the increased Crossover count did not particularly improve the GA's over all effectiveness when compared to GA 1 (39/15/2/0.1). The adapted GA 2 (39/15/6/0.1) converged to a more expensive total cost of R129,845 per day in the second test, an increase of roughly R300 per day. Surprisingly the optimal IF set generated by GA 2 (39/15/6/0.1) in the second test, consists of only two open IFs [1 and 8], yet it costs more than the first test of GA 1 (39/15/2/0.1) with four IFs [1, 3, 5 and 9]. Because of the fewer IFs in the solution, the increased cost can only be ascribed to an increased routing cost associated with the specific IF set (since the objective function only consists of IF costs and Routing costs). The LARP considered here (consisting of the benchmark problem with added IFs) therefore has a very complex solution space and is in agreement with the findings of the literature study (see Chapter 2.3 on the complexity of the LARP).

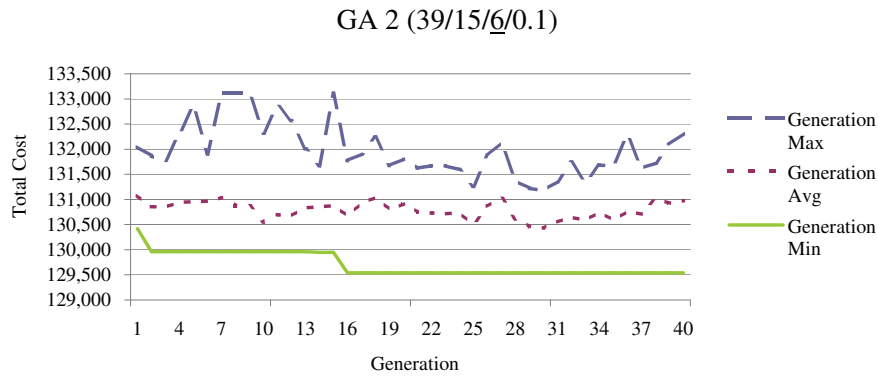


Figure 4.4: Increasing the Crossover count (Population size of 15)

- Minimum Objective Value = 129,542
- IF set proposed = [1.0.0.1.0.0.0.0.0.]
- Computation time to arrive at this solution (h) = 8.5
- Convergancy: Good
- Unstable variation: The population's average converges slightly over generations.

- Stable variation: There is an abundance of variation in the generation's worst objective functions.

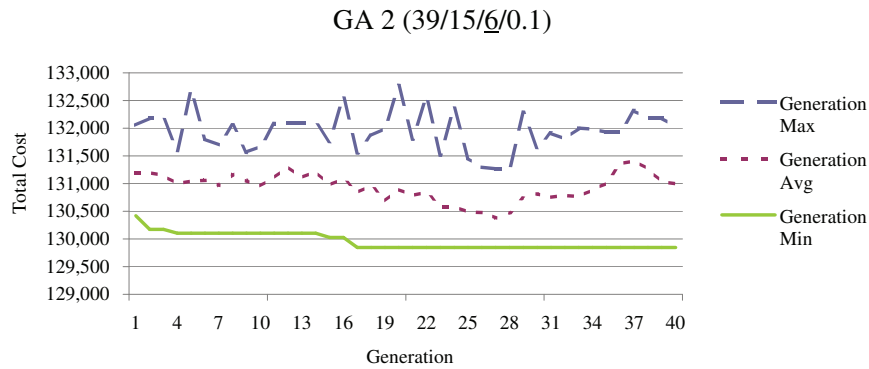


Figure 4.5: Increasing the Crossover count 2 (Population size of 15)

- Minimum Objective Value = 129,845
- IF set proposed = [1.0.0.0.0.0.0.1.0.0.]
- Computation time to arrive at this solution (h) = 8.5
- Convergancy: Good and in frequent small steps
- Unstable variation: The population's average does converge slightly over generations.
- Stable variation: There is an abundance of variation in the generation's worst objective functions.

### Reduced Mutation probability

The typically desired form of a GA has been achieved in GA 3 (39/15/2/0.05) by reducing the Mutation probability, as can be seen in Figures 4.6 and 4.7.

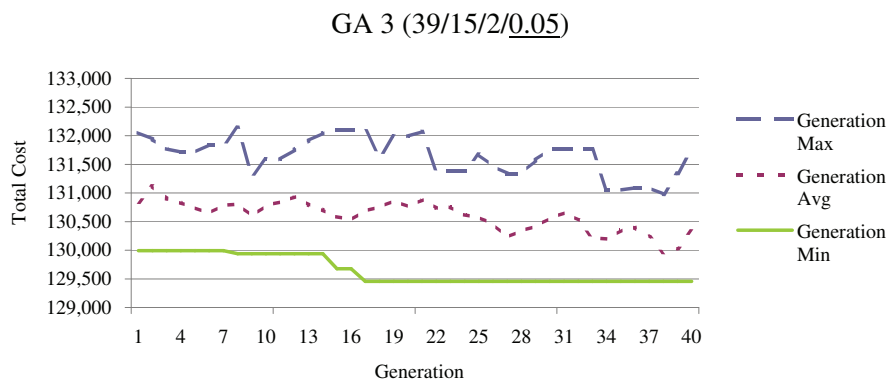


Figure 4.6: Reducing the Mutation probability (Population size of 15)

- Minimum Objective Value = 129,594
- IF set proposed = [1.0.1.0.1.0.0.0.1.0.]

- Computation time to arrive at this solution ( $h$ ) = 8.925
- Convergancy: Good
- Unstable variation: The population's average converges very well over generations.
- Stable variation: There is an abundance of variation in the generation's worst objective function and it also converges well.

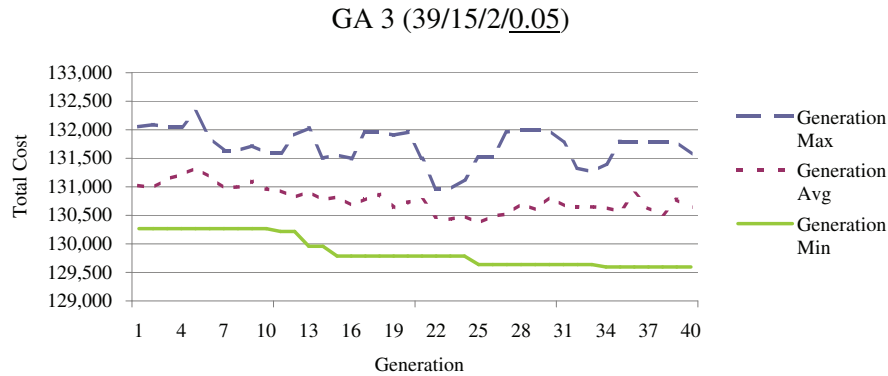


Figure 4.7: Reducing the Mutation probability 2 (Population size of 15)

- Minimum Objective Value = 129,594
- IF set proposed = [1.0.1.0.1.0.0.0.1.0.]
- Computation time to arrive at a good solution ( $h$ ) = 12.6
- Convergancy: Good
- Unstable variation: The population's average converges very well over generations.
- Stable variation: There is an abundance of variation in the generation's worst objective function and it also converges slightly.

### Increased Mutation probability

In Figure 4.8 we see that the best solution found by the GA does not converge at all. This is because the initial random population generated by the GA included this good solution and therefore the GA struggled to find a possible better one. The second test (Figure 4.8) achieved a slightly more expensive solution than the first test, but it converges very well as a whole. This was not expected since the increased Mutation probability is more likely to decrease convergence as it did in the first test. Once again the intrinsic randomness of the GA is probably the cause of this.

- Minimum Objective Value = 129,542
- IF set proposed = [1.0.0.1.0.0.0.0.0.0.]
- Computation time to arrive at a good solution ( $h$ ) = 0
- Convergancy: Immediately. This can be ascribed to the randomness in generating the initial population.
- Unstable variation: The population's average does not converge over generations.
- Stable variation: There is an abundance of variation in the generation's worst objective functions but it actually got worse over generations.

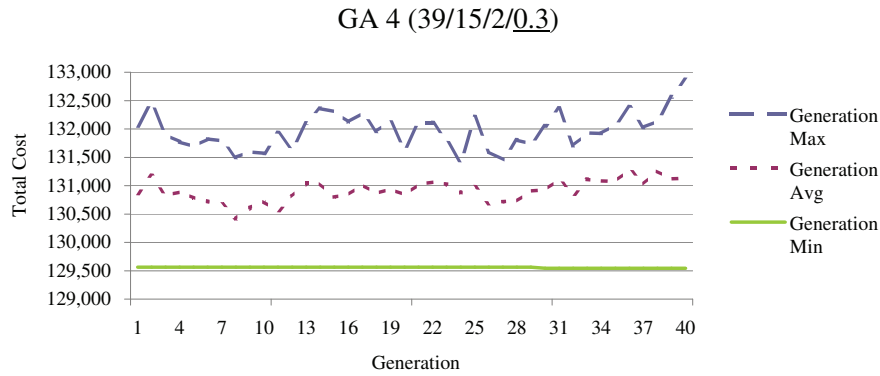


Figure 4.8: Increasing the Mutation probability (Population size of 15)

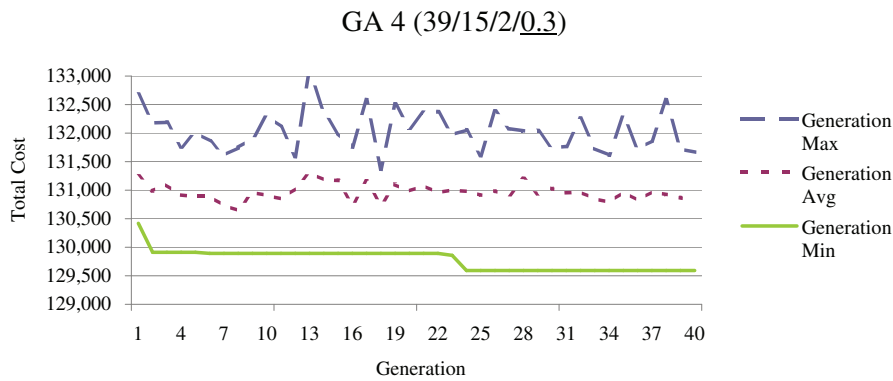


Figure 4.9: Increasing the Mutation probability 2 (Population size of 15)

- Minimum Objective Value = 129,594
- IF set proposed = [1.0.1.0.1.0.0.0.1.0.]
- Computation time to arrive at a good solution (h) = 12.6
- Convergancy: Good
- Unstable variation: The population's average converges well over generations.
- Stable variation: There is an abundance of variation in the generation's worst objective function and it also converges well.

## Conclusion

Most of the GA's converge, or converge close to the same total cost of R129,594. GA 1 seems to be the best version in terms of finding quality solutions. GA 3 has a very good overall convergence but does not get the same good solutions as GA 1.

### 4.1.4 Analysing a population size of 11

The previous results showed that no two same variants of the GA obtained similar results. There were much randomness present in the GA. To reduce the randomness, each variant of the GAs tested with a population size of 11 had the same initial population. Four runs were conducted and the average initial population was chosen (see Appendix A). Figure 4.10, 4.11, 4.12 and 4.13 depict

the effect that a smaller population size has on the computation time of the GA, when finding a good solution to the LARP.

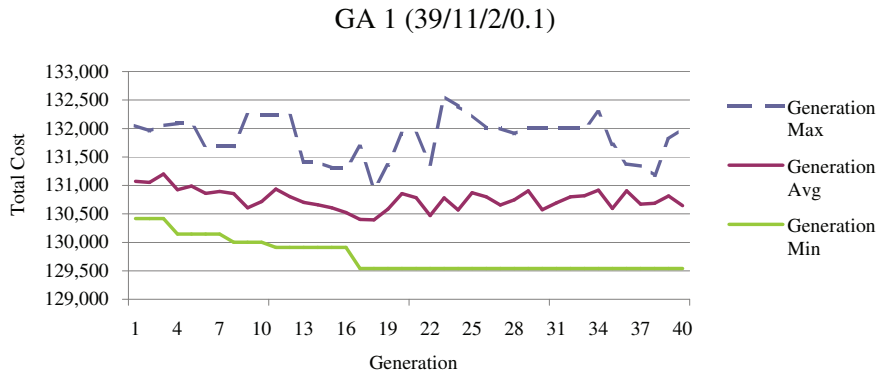


Figure 4.10: Standard GA (Population size of 11)

- Minimum Objective Value = 129,542
- IF set proposed = [1.0.0.1.0.0.0.0.0.]
- Computation time to arrive at this solution (h) = 4.675
- Convergancy: Fast
- Unstable variation: The population's average converges well over generations.
- Stable variation: There is an abundance of variation in the generation's worst objective functions, with slight convergence.

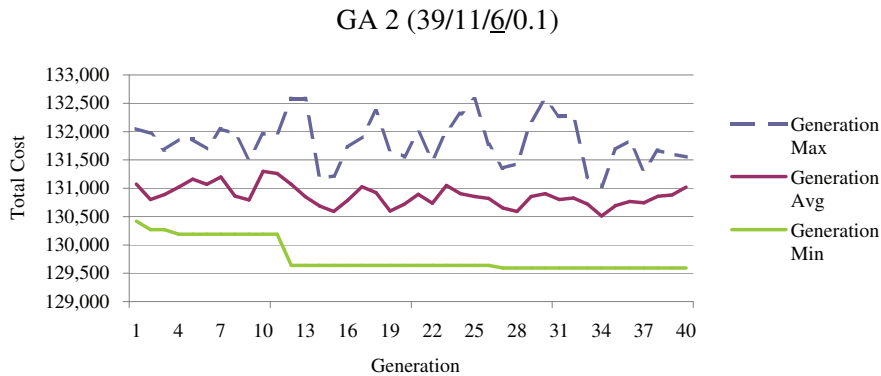


Figure 4.11: Increasing the Crossover count (Population size of 11)

- Minimum Objective Value = 129,594
- IF set proposed = [1.0.1.0.1.0.0.0.1.0.]
- Computation time to arrive at this solution (h) = 7.425
- Convergancy: Fast
- Unstable variation: The population's average does not converge over generations.

GA 3 (39/11/2/0.05)

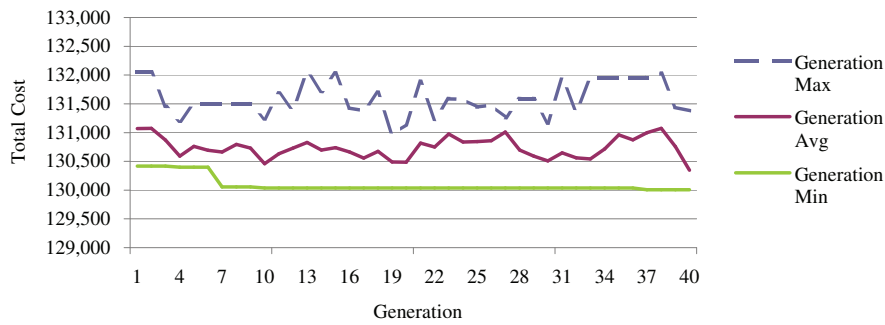


Figure 4.12: Reducing the Mutation probability (Population size of 11)

- Stable variation: There is an abundance of variation in the generation's worst objective functions, with good convergence present.
- Minimum Objective value = 130,005
- IF set proposed = [0.1.0.0.1.0.0.0.0.0.]
- Computation time to arrive at this solution (h) = 2.75
- Convergancy: Fast
- Unstable variation: The population's average converges slightly over generations.
- Stable variation: There is an abundance of variation in the generation's worst objective functions, with slight convergence.

GA 4 (39/11/2/0.3)

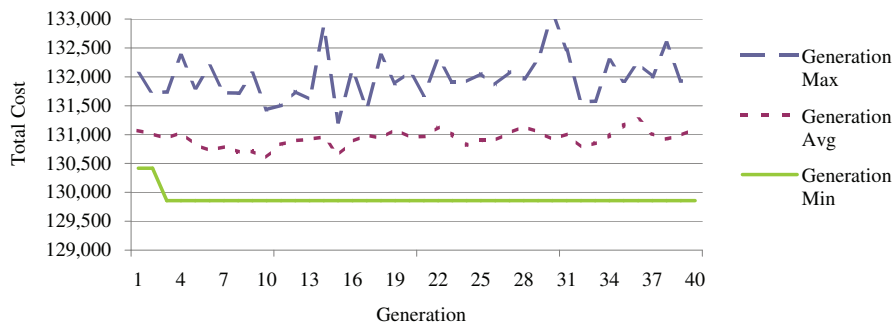


Figure 4.13: Increasing the Mutation Probability (Population size of 11)

- Minimum Objective Value = 129,857
- IF set proposed = [1.0.1.0.1.0.0.1.0.0.]
- Computation time to arrive at this solution (h) = 0.825
- Convergancy: Unusually fast
- Unstable variation: The population's average remains constant over generations.

- Stable variation: There is an abundance of variation in the generation's worst objective functions.

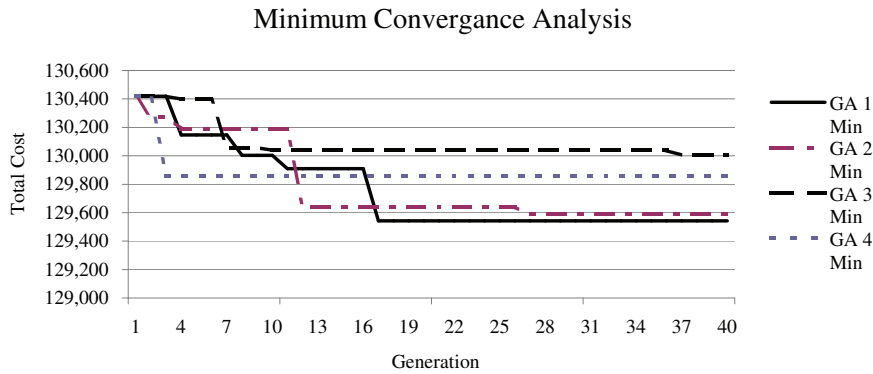


Figure 4.14: Best solution convergence (GA's with a Population size of 11)

## Conclusion

The smaller population size of 11 proves to cut the computation time drastically while still providing good solutions to the LARP. The variant GAs are all capable of finding good solutions but when looking for the best solution one should consider GA 1 although its form is not that of a natural GA. It seems the mixture of randomness and the stability which a GA provides (found in GA 1) is the reason for this. The best solution found for the specific LARP is:

- Minimum Objective value or total cost = R129,542
- IF set = [1.0.0.1.0.0.0.0.0.]
- New IF to be erected on node = [76] IF number 4
- Old IF to be shut down on node = [34] IF number 2
- Existing IF to be kept open on node = [13] IF number 1

## Chapter 5

# Shortcomings and Improvement Recommendations

Altogether over 200 hours of computation time was used to evaluate and refine the GA proposed by this project. Although the algorithm produces good solutions there is a point of concern with regards to the convergence rate of the algorithm. The populations' averages do not converge as expected.

The intrinsic randomness of the GA calls for more extensive tests for which there was not a sufficient amount of time for in this project. 200 hours of computation time did not prove to be enough to evaluate the variants of the GA with more confidence.

The effectiveness of the algorithm can be improved by starting with a very large initial population (for example twice the size of the desired population size) before the problem starts. Then evaluating the fitness and choosing the best half of the population to enter the algorithm. In so doing the algorithm starts with a much better population and might converge faster.

Fitness tournaments, whereby Roulette selected strings first compete against each other before being inserted into the child, can improve the convergence of the population as a whole. This will ensure that solutions actually improve when undergoing variation.

# Bibliography

- Coley, D. (1999). *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific Publishing.
- Cooper, L. (1963). Location-allocation problems. *Operations Research*, 11(3):331–343.
- Cooper, L. (1964). Heuristic methods for location-allocation problems. *SIAM Review*, 6(1):37–53.
- Dantzig, G. and Ramser, J. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.
- Dror, M. (2000). *Arc Routing: Theory, Solutions and Applications*. Kluwer Academic Publishers.
- Eiselt, H. (2007). Locating landfills-optimization vs. reality. *European journal of Operational Research*, 179:1040–1049.
- Gellet, B. (1976). *Introduction to operations research: A computer oriented algorithmic approach*. McGraw-Hill series in industrial engineering & management science.
- Gendreau, M., Hertz, A., and Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, 40(10):1276–1290.
- Ghiani, G., Guerriero, F., Laporte, G., and Musmanno, R. (2004). Tabu search heuristics for the arc routing problem with intermediate facilities under capacity and length restrictions. *Journal of Mathematical Modelling and Algorithms*, 3:209223.
- Ghiani, G., Importa, G., and Laporte, G. (2001). The capacitated arc routing problem with intermediate facilities. *Networks*, 37(3):134–143.
- Ghiani, G. and Laporte, G. (1999). Eulerian location problems. *Networks*, 34:291–302.
- Ghiani, G. and Laporte, G. (2001). Location-arc routing problems. *Opsearch*, 38(2):151–159.
- Golden, B. and Wong, R. (1981). Capacitated arc routing problems. *Networks*, 11(2):305–315.
- Hansen, P., Mladenovic, N., and Perez-Britos, D. (2001). Variable neighborhood decomposition search. *Journal of Heuristics*, 7(4):335–350.
- Hertz, A., Laporte, G., and Mittaz, M. (2000). A tabu search heuristic for the capacitated arc routing problem. *Operations Research*, 48(1):129–135.
- Letchford, A. and Oukil, A. (2009). Exploiting sparsity in pricing routines for the capacitated arc routing problem. *Computers & Operations Research*, 36:2320–2327.
- Levy, L. and Bodin, L. (1989). The arc oriented location routing problem. *INFOR*, 27:74–94.
- Liu, T., Jiang, Z., Chen, F., Liu, R., and Liu, S. (2009). Combined location-arc routing problems: A survey and suggestions for future research. Department of Industrial Engineering and Logistics Management Shanghai Jiao Tong University.
- Miehle, W. (1958). Link-length minimization in networks. *Operations Research*, 6(2):232–243.

- Moder, J. and Elmaghraby, S. (1978). *Handbook of operations research*. Van Nostrand Reinhold Company.
- Pang, T. (2006). *An Introduction to Computational Physics*. Cambridge University Press, 2 edition.
- Pearn, W., Assad, A., and Golden, B. (1987). Transforming arc routing into node routing problems. *Computers and Operations Research*, 14:285–288.
- Prins, C., Sevaux, M., and Sorensen, K. (2003). A genetic algorithm with population management (gajpm) for the carp. Internet.
- Rardin, R. (1997). *Optimization in Operations Research*. Prentice Hall.
- Sbihi, A. and Eglese, R. W. (2010). Combinatorial optimization and green logistics. *Ann Oper Res*, 175:159175.
- Toth, P. and Vigo, D. (2002). *The Vehicle Routing Problem*. Society of Industrial and Applied Mathematics.
- Winston, W. and Venkataramanan, M. (2003). *Introduction to Mathematical Programming*, volume One of Operations research. California: Duxbury.

## Appendix A

Chart showing four initial populations generated with respective statistics

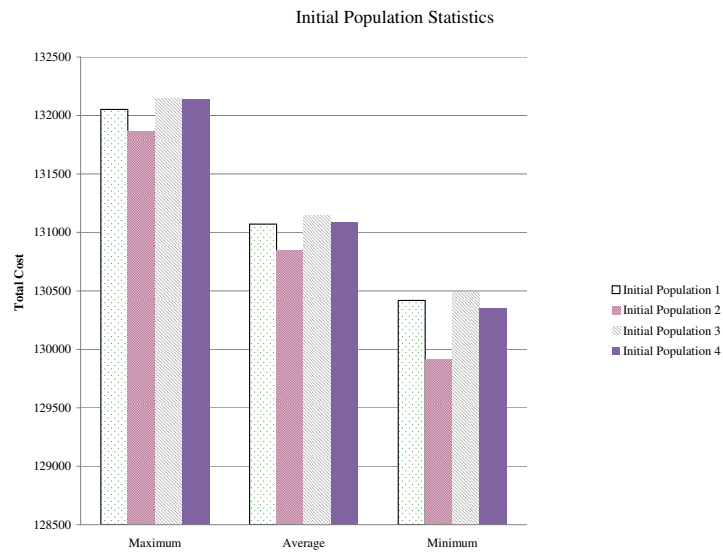


Figure 5.1: Evaluating the Initial Population

Original Population chosen(Initial Population 1):

```
[[ 1. 0. 1. 1. 0. 0. 1. 0. 0. 1.]  
[ 1. 1. 1. 1. 1. 0. 1. 1. 0. 0.]  
[ 1. 0. 1. 0. 0. 1. 0. 0. 1. 1.]  
[ 1. 0. 0. 0. 0. 1. 1. 1. 0. 0.]  
[ 0. 0. 1. 1. 0. 0. 0. 1. 0. 1.]  
[ 0. 1. 1. 1. 0. 1. 1. 0. 0. 1.]  
[ 0. 1. 1. 1. 1. 1. 1. 0. 1. 1.]  
[ 0. 0. 0. 0. 1. 1. 1. 0. 0. 1.]  
[ 1. 1. 1. 1. 0. 1. 1. 0. 0. 0.]  
[ 0. 0. 1. 1. 0. 0. 0. 0. 0. 1.]  
[ 1. 0. 1. 0. 0. 1. 0. 1. 0. 0.]]
```