Designing multimodal public transport networks using metaheuristics

Manuel Fletterman

Submitted in partial fulfillment of the requirements for the degree of

Master of Science (Industrial Engineering)

in the

Faculty of Engineering, Built Environment
and Information Technology

University of Pretoria
Pretoria

April 2008

Designing multimodal public transport networks using metaheuristics

Manuel Fletterman

Submitted in partial fulfillment of the requirements for the degree of

Master of Science (Industrial Engineering)

in the

Faculty of Engineering, Built Environment and Information Technology

University of Pretoria
Pretoria

April 2008

Abstract

The public transport system in South Africa is in a precarious state, capturing no more than 50% of the passenger market. The three public transport modes that are currently utilized—train, bus, and minibus-taxi—are competing for market share instead of complementing one another. Furthermore, most public transport networks have not been properly redesigned over the past three decades. Improvements were initiated reactively in the past: transit stops and routes were added or removed from the network when demand fluctuated. This reactive process has diminished the confidence of commuters in the public transport networks, forcing commuters to use private transport.

A proactive redesign method is needed—one that includes all the modes of public transport, and anticipates an increase in demand and rapid development in geographic areas, while ensuring good accessibility to the network. Current network design models do not include multiple modes of public transport, and are based on the geographical layout of developed cities and their particularities, which makes them unsuitable for the South African environment with its unique land use disparities.

This dissertation proposes a multimodal network design model that is capable of designing real world and large scale networks for the South African metropolitan areas. The City of Tshwane Metropolitan Municipality (CTMM) transport network area was used to develop and test the model, which consists of four components. The Geographic Information System (GIS) component has a central role in storing, manipulating, and exchanging the geographic data within the model. For the GIS the appropriate input data is identified, and a design for the geo-database is proposed. The Population Generation Algorithm (PGA) component translates the demographic data into point data representing the transit demand in the study area. The Bus Stop Placement Algorithm (BSPA) component is a metaheuristic that searches for near-optimal solutions for the placement of bus stops in the study area. A novel solution approach proposed in this dissertation uses geographic data of commuters to evaluate the bus stop placement in the study area. The Multimodal Network Design Algorithm (MNDA) component also employs a metaheuristic, enabling the design of near-optimal multimodal networks. The addition of multiple modes to the Transit Network Design Problem (TNDP) is also a novel and significant contribution. The two metaheuristic components are first tested on a test network, and subjected to a comprehensive sensitivity analysis. After identifying suitable parameter values and algorithm settings, the components are applied to the entire CTMM.

Keywords: Public transport; transport network; transport system; metaheuristic; operations research; simulated annealing; geographic information system; multimodal network design; bus stop placement; transit network design problem.

Supervisor: Dr. Johan W. Joubert

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Acknowledgements

I would like to express my sincerest thanks and appreciation to the following persons:

- My supervisor, Dr. Johan W. Joubert, for his excellent guidance and support. I am grateful for the opportunity to work with him and to learn from his effective approach to scientific research.
- The two undergraduate students, James M. Conradie and Janet Niesing, whose work formed the basis for this research project.
- My family, who gave me emotional and financial support from the Netherlands. I very much enjoyed the (almost) weekly Skype talks.
- My girlfriend, Noreen Chiku, who supported me and understood that I sometimes had to prioritize my time at the cost of our limited time together.
- All the people in the Department of Systems and Industrial Engineering at the University of Pretoria, for their kindness and support during my stay at the department.

This work is based upon research supported by the National Research Foundation (grant FA2007051100019).

Contents

List of Figures

List of Tables

List of Acronyms

BTNDP Bus Transit Network Design Problem

BSPA Bus Stop Placement Algorithm

CBD Central Business District

CTMM City of Tshwane Metropolitan Municipality

ESRI Environmental Systems Research Institute

FBNDP Feeder Bus Network Design Problem

FSM Four Step Model

GA Genetic Algorithm

GIS Geographic Information System

HFRGA Heuristic Feeder Route Generation Algorithm

HOP Hierarchical Optimization Process

ICRSGP Initial Candidate Route Set Generation Procedure

LSCP Location Set Covering Problem

MAD Mean Absolute Deviation

MATSim Multi Agent Transport Simulation

MNDA Multimodal Network Design Algorithm

MP Metaheuristic Procedure

NAP Network Analysis Procedure

NP-hard Non-deterministic Polynomial-time hard

OD Origin-Destination

PGA Population Generation Algorithm

TNDP Transit Network Design Problem

TrRP Transit Routing Problem

TrSP Transit Scheduling Problem

TS Tabu Search

SA Simulated Annealing

SASP Simulated Annealing Search Procedure

Chapter 1

Problem overview

1.1 Introduction

The public transport system in South Africa has been neglected for several decades. Most metropolitan areas have seen significant growth and dramatic redistribution of population, employment and retail centres. Although the mobility needs have increased, the public transport system has barely progressed. Statistics show that the current system only captures 50% of the passenger transport market, mainly the low-income group. For the other 50%, public transport is either inaccessible or not a viable alternative to private transport. This, together with the rapid growth of the large cities in South Africa, has contributed to an over-utilization of the capacity of the road network in those areas. The importance of improving the public transportation network has been acknowledged by the government and other stakeholders.

Most of the public transportation networks in South Africa have not been redesigned over the past three decades. If improvements were initiated, these were done reactively. New bus routes and bus stops were created where high demand was reported and, in turn, bus routes and bus stops were instantaneously removed where low utilization was reported. This process has resulted in a low level of confidence in the public transport networks, forcing commuters to use private transport. To reverse this cycle, a proactive redesign method is needed—one that anticipates an increase in demand and rapid development in geographic areas, while ensuring good accessibility to the network.

Proactive redesign of the network is very time-consuming, since it is a continuous process that requires extensive information-gathering and research; but it increases the efficiency of a network radically. Even so, the majority of major cities in first world countries use proactive models to help them improve their public transport network. The problem with these existing models is that they are based on the geographical layout of developed cities and their particularities, which makes them unsuitable for the South African environment. For example, the low-income group most in need of public transport is located on the outskirts of South African metropolitan areas, which is just the opposite of European and other developed urban environments, where the outskirts of cities are populated by people who can usually afford private transportation. Another important difference is the large paratransit sector in South Africa. This type of transport is referred to as minibus-taxis in South Africa, as it is in the remainder of this dissertation. These vehicles usually travel on certain predefined routes, but have no predefined bus stops. Everyone on the street is a potential bus stop, and communicates with an appropriate hand signal. Therefore, a new model needs to be developed that takes into account the socio-economic imbalances, previous urban planning strategies, and the large minibus-taxi sector.

As part of a larger research project that will assist transport planners to evaluate changes to the transport network, this project will focus on the development of a multi-modal network design model. The transport network area that will be used to develop and test the model is that of the City of Tshwane Metropolitan Municipality (CTMM). The CTMM is situated in Gauteng Province—the economic centre of South Africa—and covers Pretoria and its surrounding areas (see Figure 1.1). According to the census data published in 2001 by Statistics South Africa, the 2200 km^2 area of the CTMM is populated by 1.98 million people. The ownership structure of the public transport in the CTMM is non-integrated: train services are performed by the national rail company, bus operators consist of a municiple bus service and several commercial companies, and the minibus-taxis are usually privately owned and operate through taxi associations.

1.2 Problem statement

To improve the public transport system in an area such as the CTMM, which represents South Africa's disparate commuting circumstances, a new proactive network design model needs to be developed. The model needs to incorporate the three modes of public transport, namely train, bus, and minibus-taxi, and should ensure that the three modes complement one another instead of competing for market share (as currently happens), while ensuring that none of the stakeholders involved is negatively affected. Even so, once

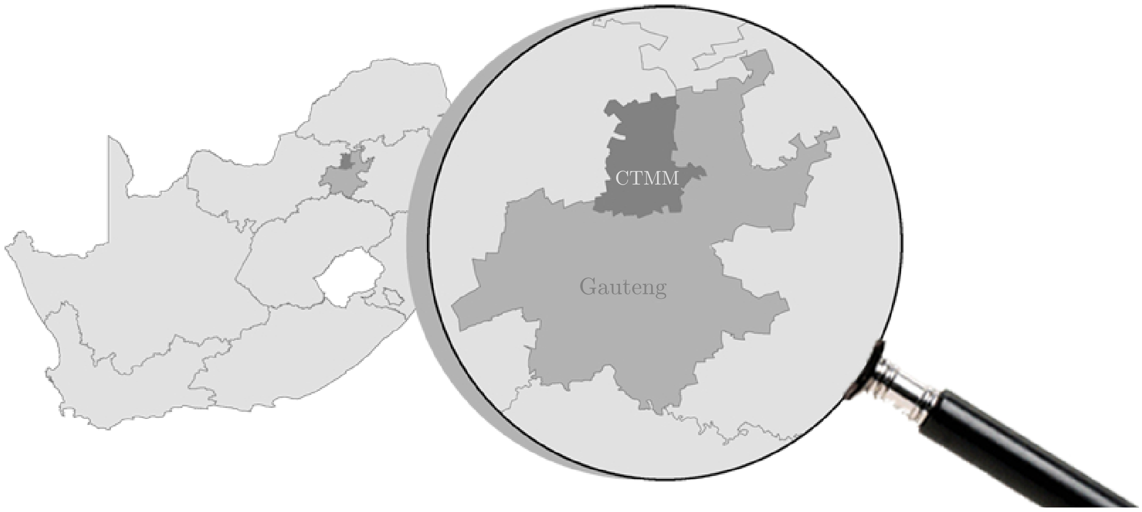


Figure 1.1: Location of the CTMM in Gauteng Province, South Africa

an improved network is implemented, the increase of market share of the whole public transport sector will benefit all the involved stakeholders.

The new model needs to be able to initiate a complete re-design of the bus stops in the network based on geographic data, instead of small adjustments on the current network. The current status of the public transport system demands such drastic measures. The advantage of such a model is the ability to integrate the various owned network services, and that the design of a network can be focused on certain demographic target groups, which will create a model that can easily adjust its design as the public transport sector increases its market share. Some bus stops, however, cannot be easily moved—for example, a bus stop next to a train station—and therefore, the model should be able to specify some bus stops as fixed. Placing bus stops in a network is a multi-objective problem, since a balance need to be found between the distance commuters must walk to a bus stop and the number of bus stops in the network.

After the placement of the bus stops, the route design can be initiated. The problem of designing a public transport network is known in literature as a Transit Network Design Problem (TNDP). Zhao (2006) classifies the TNDP as a Non-deterministic Polynomial-time hard (NP-hard) mixed combinatorial optimization problem. A problem is classified as 'NP-hard' if solving such a problem in polynomial time would make it possible to solve all the problems of the class NP in polynomial time. To date there exists no proof as to whether or not P = NP is true. A 'mixed' problem is one that contains both discrete and continuous variables. A 'combinatorial' problem usually refers to an integer optimization problem where the unknown variable set (called combinatorial set) consists of all feasible integer subsets of a larger base integer set. In order to determine a realistic network for a large area such as the CTMM, the solution space of the problem is too large and complex to be solved with traditional mathematical optimization techniques and therefore needs to be solved by adopting a search procedure that has the ability to prevent itself from getting trapped in local optima, while searching through the large solution space. The problem is also multi-objective, due to the overall complexity that requires several criteria when measuring the quality of a certain proposed solution.

From the previously discussed matters, the following research question is derived:

> *"How can a fully-operational network design model be built that will produce an optimized public transport network for South African metropolitan areas, taking the country's particulars into account?"*

1.3 Research design

The aim of the network design model is to produce an optimized network for the three modes of public transport. To achieve this, a model is developed that consists of four main components. Figure 1.2 shows the components, steps, and data exchange within the model. Each component, and its role within the model, is briefly discussed below.

Geographic Information System (GIS) The GIS can be divided into a geodatabase, which stores all the special data, and the GIS software, which enables access, manipulation and management of the data in the geodatabase. The GIS, which is involved in all the five steps of the model, has a central role in storing, manipulating, and exchanging data within the model, as well as presenting the final results. In step 1 of the model, the input data, such as the road network, zoning, sensus, and survey data, is loaded into the geodatabase using the GIS software. The data is manipulated in order to prepare it for the use of the three algorithms. In step 4, the GIS software is used to produce an Origin-Destination (OD) matrix with calculated distances between every bus stop, taking into consideration the realistic road network with one-way streets etc. In step 5 the results of the model are presented using a GIS map of the area.

Population Generation Algorithm (PGA) The purpose of the PGA is to trans-

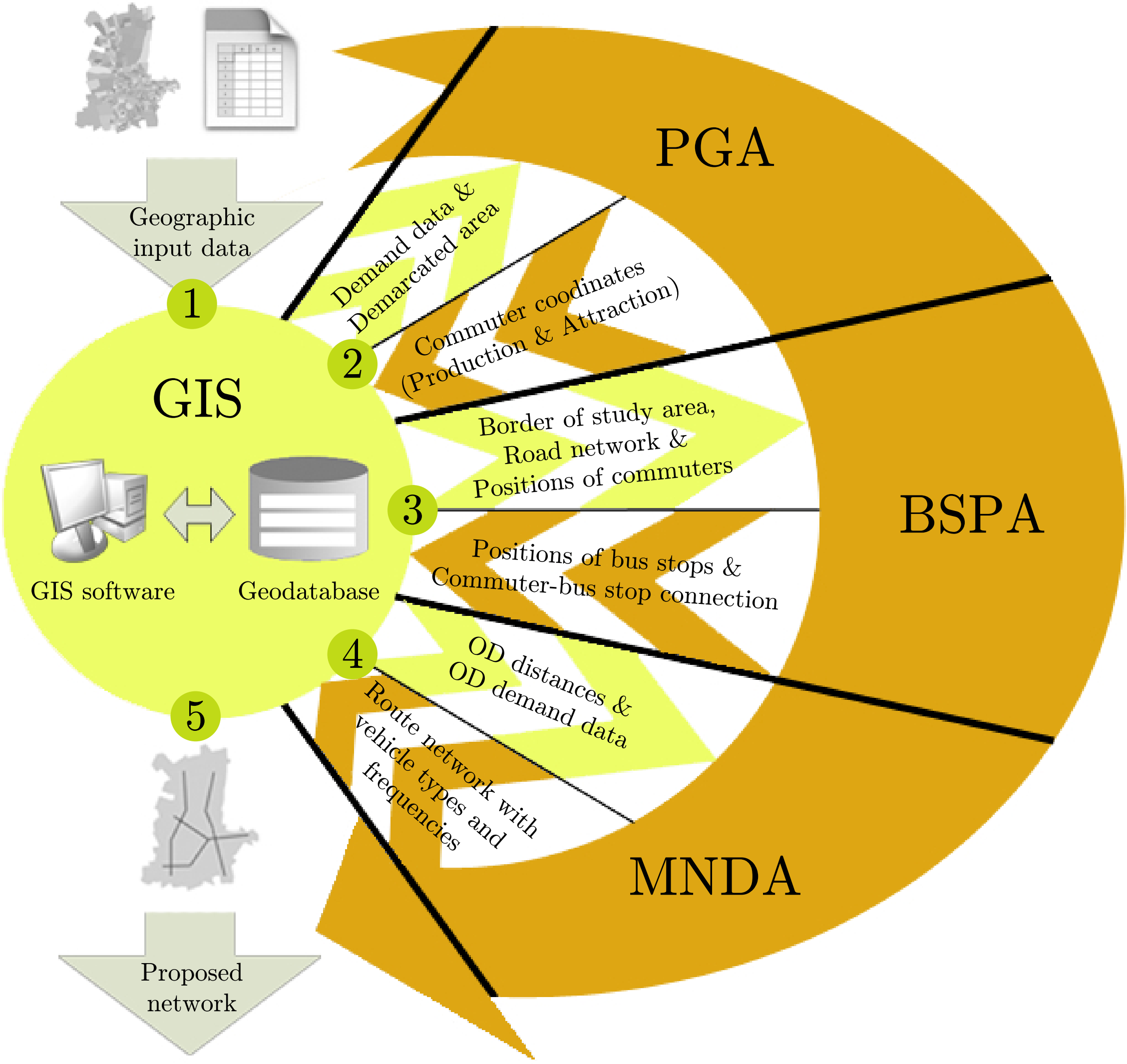


Figure 1.2: The components, steps, and data exchange within the proposed model

late the demographic data stored in the GIS to point data representing the transit demand in the study area. For this project, a simple algorithm is designed that randomly distributes the demand within demarcated areas, such as transportation zones or political wards. The PGA is used in step 2 of the model. The result of the algorithm is stored in the geodatabase.

Bus Stop Placement Algorithm (BSPA) For the placement of the bus stops in the study area, the BSPA is developed. In step 3 of the model, this algorithm seeks to find the optimal number of bus stops and their positions in the study area, based on the positions of the commuters provided by the PGA. The positions of the bus stops will be matched to the existing road network to ensure a realistic bus stop network.

The positions of the bus stops and the data linking each commuter to its assigned bus stop is then sent back to the geodatabase.

Multimodal Network Design Algorithm (MNDA) In step 4 of the model, the MNDA uses data produced by all the other components in order to find an optimized multimodal network. A solution space is created, from which the algorithm seeks to find the optimal combination of routes.

1.4 Research methodology

The main development effort is needed for the BSPA and MNDA. These are the two optimization algorithms that form the core of the model. Both are based on previous studies—Freeman and Minkwitz (2002) and Fan and Machemehl (2006b) respectively—but changes to the design are made in order to make the model applicable to the South African environment. The BSPA needs to find an optimal solution for the placement of bus stops in a real world scenario based on the geographic data of commuters in the study area—something that was not found in the literature. Therefore, other fields of research concerned with the problem of optimal placement are investigated in order to find an appropriate solution approach.

The MNDA requires the addition of multiple vehicle types to the TNDP. Several solution approaches to the TNDP are presented in the literature, but none provide the option to use multiple vehicle types. For the development of the MNDA, one of these approaches found in the literature is selected and used as a guideline for the development of a new approach that includes the use of multiple vehicle types.

Both the BSPA and MNDA are tested on a small test area/network to evaluate their performance, before they are used on the whole CTMM area.

For the GIS component, the gathering of accurate input data and the design of the geodatabase are the main concerns. For the output of the model to have any value for a real world scenario, comprehensive and up-to-date input data is required. The latest data is collected from government, municipal departments, and commercial companies. The design of the geodatabase is structured in such a way that all the required data is grouped according to its function in the model.

For the BSPA to find a near-optimal solution, the distances between bus stops and commuters need to be calculated. Therefore, the transit demand needs to be represented as points within the study area. The usual format in which demographic data on a certain area is presented is by dividing the area into sub-areas and providing the appropriate values for these sub-areas. The role of the PGA is to translate these per-sub-area values into actual point data. For the purpose of this project, the PGA can be a fairly simple algorithm that randomly distributes the transit demand in each sub-area.

The three algorithms are developed using the software package *MATLAB® R2007b*. Additional toolboxes can be added to the software package, which makes the programming language useful for a large variety of applications. For this project the *Mapping Toolbox™* is used to enable the algorithms to read the geographic data from the geodatabase, and the *Distributed Computing Toolbox™* for the purpose of sending the jobs to a multi-processor cluster called *Velocity*[1]. The software used to interface with the GIS is called *ArcGIS®*. This product is developed by the Environmental Systems Research Institute (ESRI), which is the world leader in GIS modelling and mapping software. Most of the local governments in South Africa also use this software to manage their geographic data.

1.5 Document structure

This chapter has formulated the problem faced in this project and proposed a model to address this problem. In Chapter 2 the literature review is given. Serveral metaheuristic approaches are examined, and previous studies on the bus stop placement problem and the TNDP are discussed. The chapter concludes with previous studies regarding the integration of GIS into optimization problems. In the following chapters the components of the model are discussed. Chapter 3 presents the development of both the GIS and the PGA components. The input data required by the GIS, the design of the geodatabase, and the structure of the PGA are given. Chapter 4 focuses on the development of the BSPA. The model formulation and the solution approach are given. After this the results of a test of the algorithm on a test area is presented. Chapter 5 follows the same structure as Chapter 4, but focuses on the MNDA. The results of the model on the CTMM study area are presented in Chapter 6, where each of the five steps of the model is discussed in detail. Finally the conclusions and recommendations are given in Chapter 7.

[1]The Velocity cluster is a High Performance Computing (HPC) cluster at the University of Pretoria that consists of 1 cluster master (Dell 2850, 2 x 3.4 GHz Intel Xeon, 2048 MB RAM) and 24 cluster nodes (Dell 750, 1 x 3.0 GHz Intel Pentium IV, 1024 MB RAM). More details can be found at: http://www.tlug.org.za/wiki/index.php/The_UP_HPC_cluster.

Chapter 2

Literature review

For the development of the proposed model, knowledge from several research fields is needed. Two components of the model require a metaheuristic approach in order to find a near-optimal solution within the large solution space of the problem. Therefore, three regularly used metaheuristic approaches are investigated. In order to find an appropriate solution approach for the Bus Stop Placement Algorithm (BSPA), previous literature is examined concerning the optimal placement of bus stops in a network and of base stations in a new communication network. For the development of the Multimodal Network Design Algorithm (MNDA), previous studies on the Transit Network Design Problem (TNDP) are discussed. The different multi-objective approaches to the problem are presented, the Bus Transit Network Design Problem (BTNDP) is discussed (which focuses on the bus component of the TNDP), and the Feeder Bus Network Design Problem (FBNDP) is discussed (which focuses on the integration of bus and train networks). To validate the choice to include the Geographic Information System (GIS) component in the model, previous research on the integration of GIS into optimization problems is investigated.

2.1 Metaheuristic approaches

Many real-world scenarios cannot be solved realistically by traditional optimization techniques, or else would require extensive computational resources. Therefore, metaheuristics are used to find near-optimal (possibly optimal) solutions. A metaheuristic searches through the solution space with the ability to prevent itself from getting trapped in local optima. The necessity of this is illustrated in Figure 2.1, where a simplified two dimensional solution space of a fictitious optimization problem is drawn. The solution space consists of many local optimal solutions, which are all situated at the bottom of a crevice (for a minimization problem). As the search for an optimal solution progresses, the first local optimum that is encountered could be an extremely bad solution compared with the global optimum. Therefore the solution should not be adopted right away: the metaheuristic should rather continue the search in pursuit for the global optimum. This involves 'climbing out' of the local optimum by accepting poorer solutions. Three metaheuristics that are regularly found in the literature are Simulated Annealing (SA), Genetic Algorithm (GA), and Tabu Search (TS). Each will be discussed briefly in the following sections.

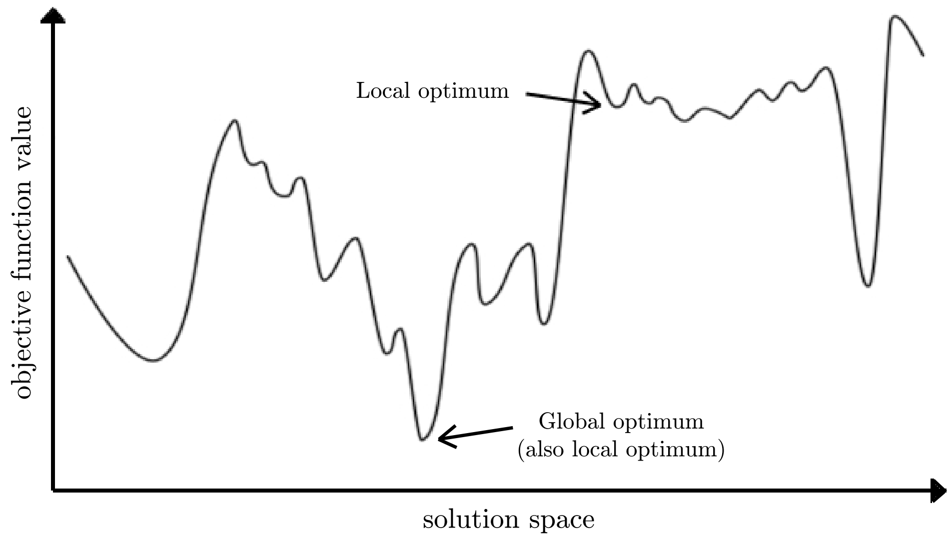


Figure 2.1: Global and local optimum in the solution space. Adopted from Fan (2004).

2.1.1 Simulated Annealing

SA was inspired by the annealing process in metallurgy. This is a technique that involves the heating and controlled cooling of a material, in order to increase the size of its crystals and reduce the number of defects. The initial heat causes the atoms to become loose from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy. The slow cooling process then gives the atoms the chance to find configurations with lower internal energy than their initial ones.

An analogy to SA may be drawn. The initial randomly generated solution of the algorithm will have a high objective function value (representing the internal energy). During each step of the SA process, the algorithm replaces the current solution with one close to it. Improving solutions will always be accepted, but non-improving solutions can also be accepted. The probability of accepting non-improving solutions depends on the 'temperature' of the SA cooling schedule and the amount by which the solution deteriorates the objective function value, as indicated in (2.1). The temperature, denoted by T, is slowly decreased over time, decreasing the probability of accepting non-improving solutions (representing the slow cooling process). The initial large probability of accepting non-improving solutions allows the algorithm to escape local optimum and explore the solution space, while the slow decrease of the probability allows the algorithm eventually to find a solution with a very low objective function value—hopefully the global optimum.

The SA algorithm is guided by its cooling schedule. The value of T depends on the initial temperature T_0, the temperature reduction factor T_{reduc}, the number of iterations performed at each temperature T_{count} and the minimum temperature needed to be reached before the algorithm stops, T_{min}. Figure 2.2(a) shows a flow diagram of the basic SA algorithm.

$$P_{\text{accept}} = e^{\frac{\Delta obj}{T}} \tag{2.1}$$

where:

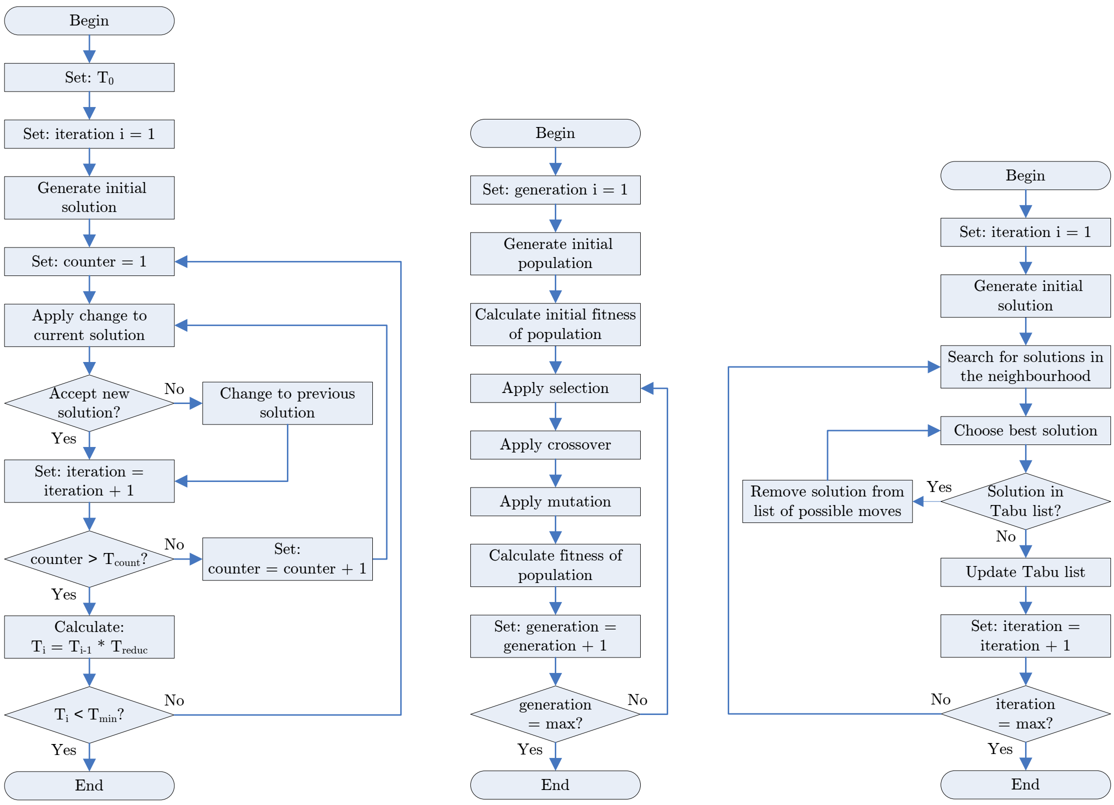
Δobj	$\triangleq$	The difference between the objective function value of the previous and current solution. ($obj_{i-1} - obj_i$ for minimization problems and $obj_i - obj_{i-1}$ for maximization problems)
T	$\triangleq$	The current temperature value of the SA cooling schedule.

2.1.2 Genetic Algorithm

GAs are search algorithms based on concepts of natural selection and natural genetics. A GA consists of a set of chromosomes, with each chromosome representing a binary gene string that corresponds with a certain solution. Initially a certain population is randomly generated. Each member of the population is evaluated according to its fitness, i.e. its objective function value. Members with higher fitness values have higher probabilities of being selected to take part in the reproduction process. Reproduction generates a new population by performing three distinct genetic operations—namely, selection, crossover, and mutation. The process continues for a fixed number of generations or until a termination trigger is received. Figure 2.2(b) shows a flow diagram of the basic GA. The GA method differs from other search methods in that it searches among a population of points and works with a coding of parameters set, rather than with parameter values themselves (Pattnaik et al., 1998).

2.1.3 Tabu Search

TS enhances the performance of a local search method by using memory structures. The TS explores the solution space by moving from one solution to another with the best objective function value in its neighbourhood, even if the value is worse than the current solution. Previous selected solutions are kept up-to-date in a list and marked as 'taboo' (hence the name, 'tabu search') to avoid cycling. This process is repeated for a certain number of iterations or until a stopping criterion is met. Figure 2.2(c) shows a flow diagram of the basic TS algorithm.



(a) SA (b) GA (c) TS

Figure 2.2: Flow diagram of the basic SA, GA, and TS algorithms

The performance of each metaheuristic depends on the characteristics of the problem, such as the type of problem and the size of the solution space. Therefore, in order to determine which of the metaheuristic approaches is the most appropriate for the development of the BSPA and the MNDA, previous literature is examined.

2.2 The placement of bus stops

The placement of bus stops has a large effect on the performance of a transportation system. Schöbel (2005) points out that the best distribution of bus stops is not obvious, since even from the commuter's point of view, the following two conflicting effects occur: 1) many stops are advantageous, since they increase accessibility; but 2) each additional stop increases the transportation time, since it decreases the average speed of a bus. An optimal solution will balance the two objectives, while satisfying the constraints.

In the study of Murray (2001) the optimal number of bus stops in a network is investigated. The model uses a strategic approach for measuring the degree of redundancy and inefficiency in bus stop coverage for an existing public transportation network, using the Location Set Covering Problem (LSCP). The objective of the LSCP is to minimize the number of bus stops needed to provide complete access coverage to the whole service area. As input to the model, a certain value is chosen for the minimal allowed distance between a service area and a bus stop. In a model using circles around bus stops to indicate their coverage area, this would translate into a maximum allowed radius of the circles. The evaluation of access coverage is structured in a GIS, which enables real time user interactive exploratory spatial data analysis.

All the studies found in the literature on the optimal placement of bus stops built on existing networks, either by adjusting the spacing between bus stops or by evaluating their service coverage. None tried to build a new network based on the geographic data of commuters in the study area. Therefore, other fields of research are investigated, such as the development of new communication networks, since the placement of bus stops in a new transit network can be easily compared to the placement of base stations in a new communication network.

The study of Hao et al. (1997) describes the Hierarchical Optimization Process (HOP) used for designing a cellular mobile system. The model consists of three interactive levels: 1) determination of number of cells and average cell range; 2) selection of best cell site location and determination of optimal cell sizes, using an SA approach; and 3) detailed planning and accurate cost estimation. The first two levels of the model could be useful for developing the BSPA.

The model by Freeman and Minkwitz (2002) uses a predetermined number of base stations on a fixed area of land as an input. Circles, representing the coverage area of the different base stations, are randomly placed within the area of land. The model then starts to move the circles around in order to maximize the total coverage area, whilst minimizing the overlap of different service areas. The model can easily be adjusted and extended to meet the requirements of a model for the placement of bus stops—e.g. using a variable number of bus stops and a variable radius of the circles.

The positions of the bus stops are part of the basic representation of the network that needs to be solved by the MNDA. Once the optimal number of bus stops and their positions are determined by the BSPA, the MNDA uses them to design an optimal multimodal network.

2.3 The transit network design problem

The TNDP has been extensively covered in the literature. Several approaches have been used to solve this multi-objective problem. The different formulations of the objective function in previous literature are therefore investigated. Further investigation on the TNDP is done by reviewing previous literature on the Bus Transit Network Design Problem (BT-NDP) that focuses on the design of networks solely for buses, and the Feeder Bus Network Design Problem (FBNDP) that focuses on incorporating trains into the network by creating feeder bus lines that connect to train stations.

2.3.1 Formulations of the objective function

Optimization of problems with multi-objectives deals with the process of simultaneously optimizing two or more conflicting objectives. A solution is sought for which each objective has been optimized to the extent that, if you tried to optimize it any further, the other objective(s) would suffer. In the literature, the objective function has been formulated in many different ways. In the work of Van Nes and Bovy (2000) six formulations are investigated and tested to determine which is best suited for the TNDP. Amongst the objectives tested are:

Minimum total travel time Travel time typically consists of three components: waiting time, in-vehicle time, and transfer time. This objective is completely focused on the commuter perceptive. Therefore it is often used in combination with an operational cost constraint to ensure that the operational costs stay within a certain range. This approach was followed by Lampkin and Saalmans (1967) and Lee and Vuchic (2005).

Maximum cost effectiveness A formulation especially suited to the operators point of view. The objective is defined as the ratio of the total revenues and the operational costs. In the literature, this objective does not seem to be widely used.

Maximum profit Also from an operator's point of view, as the objective is aimed solely at producing profitable networks. The objective is defined as the total revenues reduced by the operational costs.

Maximum total passengers Produces the most friendly networks for commuters. An important component in this objective is to reduce the total travel time.

Minimum total cost The commuter and the operator are both taken into account, which enables the production of truly sustainable networks. In the literature, this objective is by far the most popular. Examples of studies using this objective are Ceder and Wilson (1986), Fan and Machemehl (2006a,b), Tom and Mohan (2003), and Ngamchai and Lovell (2003).

For the purpose of this project the 'minimum total cost' objective is also the most appropriate one. Generally the objective consists of three components: user cost, operator cost and unsatisfied demand.

2.3.2 The bus transit network design problem

The BTNDP has received quite some attention in the literature recently. Fan and Machemehl (2006a) indicate that the approaches used can be classified into three main categories: 1) practical guidelines and *ad hoc* procedures; 2) analytical optimization models for idealized situations; and 3) metaheuristic approaches for more practical problems.

Ceder and Wilson (1986) break the process of bus planning down into five sequential steps: 1) network design; 2) setting frequencies; 3) timetable development; 4) bus scheduling; and 5) driver scheduling. In most studies the BTNDP focuses on the first two steps of this process.

Chakroborty (2003) proposes to divide the BTNDP into two separate problems: the Transit Routing Problem (TrRP) and the Transit Scheduling Problem (TrSP). The problems are solved sequentially by using GAs.

Fan and Machemehl (2004, 2006a,b) propose a solution framework that consists of three main components: 1) an Initial Candidate Route Set Generation Procedure (ICRSGP) that generates all feasible routes; 2) a Network Analysis Procedure (NAP) that assigns transit trips, determines service frequencies, and computes performance measures; and 3) a Metaheuristic Procedure (MP) that combines these two parts, as it guides the candidate solution generation process and selects an optimal set of routes from the huge solution space.

In the study of Fan and Machemehl (2006b), in which an SA metaheuristic is used for the MP, the following is stated:

> "Numerical results show that the proposed SA algorithm based model seems to outperform the GA in most cases, suggesting that compared to the GA, the SA is at least as good—and potentially better—a candidate solution approach for the BTNDP".

2.3.3 The feeder bus network design problem

The integration of a rail line through feeder bus routes into the City of Tshwane Metropolitan Municipality (CTMM) transport system would greatly improve passengers' travel possibilities. Kuan et al. (2004) offer a solution approach to the FBNDP by using an SA and TS. The TS approach consists of two versions: basic TS and TS combined with an intensification strategy. The latter shows the best results overall in the comparison of the three approaches for the 20 test problems that were used, but also requires on average the longest computational times. The SA algorithm is fast and offers reasonably good solutions.

Shrivastava and Dhingra (2001) developed a model for the operational integration of suburban railway stations and public buses. In the study, a heuristic algorithm is proposed for the development of feeder routes to railway stations. Two railway stations in Mumbai (India) were taken as study areas. The proposed Heuristic Feeder Route Generation Algorithm (HFRGA) is heavily guided by the demand matrix, as it gives priority to nodes having higher demands. The scheduling part of the model is presented in Shrivastava and Dhingra (2002). A GA approach is used for optimization, in which penalties are given if a certain amount of transfer time is exceeded, for unsatisfied demand and if the loading capacity of a bus is outside of a certain predefined range.

In the work of Shrivastava and O'Mahony (2006) the FBNDP is solved by simultaneously developing feeder routes and the schedules that lead to coordination between the train and bus schedules, using a GA approach. The results of the proposed model indicate improved load factors on developed routes, and also a considerably improved overall load factor as compared to the authors' earlier model. This study also shows that the main issue with the integration of the feeder routes into the BTNDP comes down to the synchronization of the train and bus schedules. In this case the TrRP and TrSP need to be solved simultaneously.

2.4 Optimization through integration of a GIS

GIS serves as a repository of geographic information and enables spatial manipulations and database management (Jha et al., 2001). All the input data of the proposed model in this report has a geographic element to it, which makes GIS an excellent tool to manage this data. With the right GIS software, such as the *ArcGIS®* package, it is possible to use data stored in a large variety of formats—e.g. spreadsheets, text files in comma-separated value, and image files. This data can then be stored in one standard GIS data format. The *ArcGIS®* package offers the possibility of storing the data in a so-called *geodatabase*, which enables the structural arrangement of data, the application of rules and relationships to the data, the definition of geometric relational models (e.g. networks), the maintenance of data integrity, etc.

Several studies have investigated the advantages of combining GIS and search algorithms for solving optimization problems. Li and Yeh (2005) demonstrate in their study that metaheuristics can be used with GIS to solve spatial decision problems effectively. For their optimal location search a GA approach is used, integrated with GIS, to retrieve spatial data and present the output of the algorithm.

Tat and Tao (2003) state in their study that "integrating the GIS and the GA for the highway alignment problem allows one to take advantage of the strength of each technique in the search for optimal or nearly optimal highway alignments". The proposed methodology uses a continuous and automated exchange of data between the GA and the GIS. For each iteration, the cost evaluation is performed by the GIS and the optimization is done by the GA.

For both the studies of Jha et al. (2001) and Li and Yeh (2005), *ArcGIS®* was used as the software tool to manage the GIS. The study of Tat and Tao (2003) does not specify which software tool was used.

2.5 Conclusion

The similarity between allocating base stations and allocating bus stops in a new network makes it possible to use the model developed by Freeman and Minkwitz (2002). The model will need to be adjusted so that it can vary the number of bus stops and the size of the service areas. The key to the problem will be to define an appropriate objective function that balances the two conflicting objectives: to increase accessibility and decrease travel time.

The framework proposed by Fan and Machemehl (2006b), which uses an SA algorithm to search for an optimal solution, seems to be the most appropriate solution approach to solve the BTNDP. The framework consists of three main components: the ICRSGP, the NAP, and the SA algorithm-based metaheuristic implementation model.

The integration of trains into the model can be done by synchronizing train and bus schedules, as indicated by Shrivastava and O'Mahony (2006). This is a viable option for the model, since the proposed solution approach to the BTNDP solves the routing and scheduling problem simultaneously.

Previous studies show that integration of GIS into a solution approach to largely geographically orientated optimization problems provides a tremendous advantage, as it enables manipulation and good management of spatial data. The software package *ArcGIS®* has powerful data manipulation and storage capabilities, and thus will be used as the GIS software for the model.

The literature review presented here has provided the fundamental building blocks for the design of the model's components. In the following chapters the development of these components is discussed in more detail. The next chapter focuses on the GIS and Population Generation Algorithm (PGA) components.

Chapter 3

The Geographic Information System and Population Generation Algorithm

The GIS and PGA are both involved with the preparation of the input data for the BSPA and MNDA. The main development effort for the GIS is to collect all the required geographic data, manipulate it, and store it in an organized structure, so that the algorithms can access and use it. The geographic input data is identified and, in order to organize this data structurally, a design for the geodatabase is proposed. The three algorithms of the model interface with the GIS to retrieve their input data and store their results; therefore an overview is given of the interaction of the algorithms with the components in the geodatabase. For the development of the PGA, the required data from the GIS is identified and the structure of the algorithm presented.

For the design of the GIS and PGA, two approaches can be followed. The first approach uses demand data from an already existing demand generation model used by the Tshwane Transport Authority. The second approach involves the creation of a new demand generation model that uses survey, census, and land use data to generate the demand data according to its own defined rules. It is assumed that the second approach will place the commuters in more accurate positions in the study area. However, this approach requires a comprehensive study in demand generation, and a more sophisticated design of the PGA component. For the purpose of this project the first approach is followed. However, to accommodate future extension of the model proposed in this dissertation, the design of the GIS component according to the second approach is provided in Appendix A.

3.1 The Geographic Information System

GIS data represents real world objects—such as roads, parcels, and elevation—with digital data. Two main methods are used to store data in a GIS: as a vector, for discrete objects such as roads; or as a raster, for continuous fields such as elevation. Our focus is on the vector method, since all the input data used in this project can best be represented as vector data. In a GIS the vector data representing the real world objects are referred to as features, and are represented as points, lines, or polygons. The information that describes these features—such as road names, parcel ownership, and land use type—is stored in tables that are linked to the features. In the tables, each row represents a feature and each column an attribute. Features that represent the same kind of objects and share the same attributes are stored together in a feature class.

GIS data is widely used by governmental institutions and commercial companies involved with the planning and development of the public domain. The first step in developing the GIS is to determine exactly what kind of input data is needed by the model.

3.1.1 Input data

The requirements of the input data of the GIS are largely dependent on the required input data of the algorithms.

The PGA component needs demand data per demarcated area, together with the borders of these demarcated areas. The demand data is provided by the Tshwane Transport Authority, and is generated with a program called *EMME/2*®. This program uses the Four Step Model (FSM), which is a widely used model to generate demand data. This demand data is calculated per transportation zone, which divides the CTMM into 574 demarcated areas. The GIS data representing these transportation zones was provided by the Tshwane Municipality for this project.

The BSPA component needs the data produced by the PGA, data representing the border of the study area, and data representing the transport network. The border of the study area is produced by manipulating the transportation zones with the GIS software. The transport network is provided by a commercial company specialized in supplying GIS data.

The MNDA needs two Origin-Destination (OD) matrices: one that contains the distances between the stops in the network, and one that contains the demand between the stops. The distances between the stops are calculated with the GIS software, using the positions of the bus stops produced by the BSPA and the transport network. The demand between the stops is generated according to the assignment of commuters to the bus stops as provided by the BSPA.

An important step in building the GIS component is the design of the geodatabase. The geodatabase stores all the geographic data needed by the algorithms, as well as the eventual results of the algorithms.

3.1.2 Geodatabase design

For the design of the geodatabase, the 10-step approach proposed by Arctur and Zeiler (2004) is followed. The process moves from a conceptual design to a logical design, and eventually to a physical design.

The initial steps of the process help to identify and characterize the thematic layers, which are collections of common features—such as a road network or a collection of parcel boundaries. The thematic layers identified for this project are based on the requirements of the algorithms, and are characterized by the developing specifications for the representation of the layers in the physical database. Figure 3.1 shows the identified thematic layers for the design of the geodatabase.

With the use of the descriptions of the thematic layers, the representations are then grouped and modelled using feature datasets, feature classes, relationship classes, rules, and domains. Each of these geodatabase elements is briefly discussed below.

Feature class A collection of geographic features sharing the same geometry type (such as point, line, or polygon), the same attributes, and the same spatial reference (features that share a coordinate system and fall within a common geographic area).

Feature dataset A collection of feature classes stored together, sharing the same spatial reference. Feature classes with different geometry types can be stored together in a feature dataset.

Relationship class An item that stores information about a relationship. This relationship can be between feature classes, feature classes and tables, or tables.

Spatial rules These are used to model how features share geometry with other features. Examples of spatial rules are topologies and networks. A topology defines

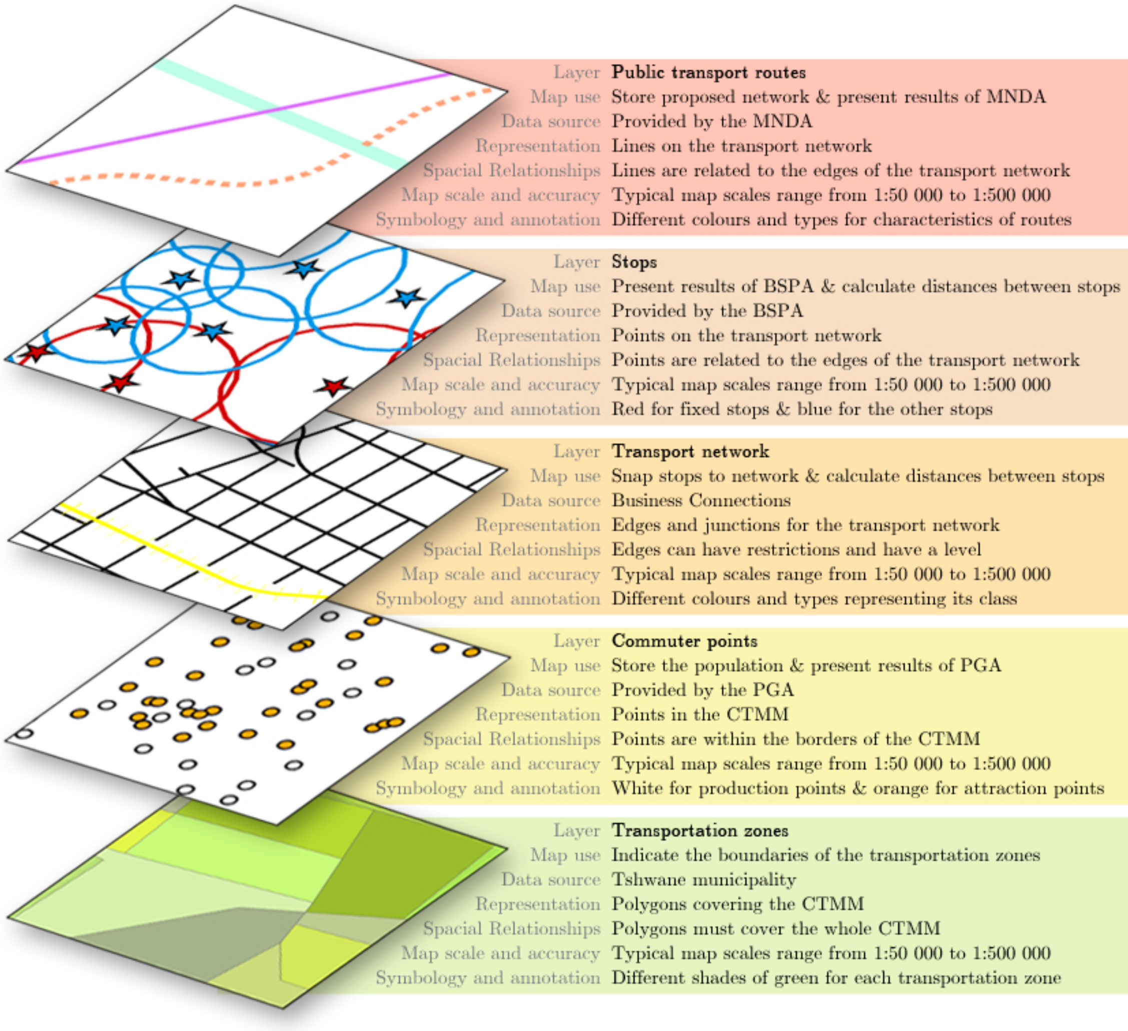


Figure 3.1: Thematic layers identified for the design of the geodatabase

and enforces data integrity rules, such as avoiding the overlap of features. A network consists of an interconnected set of edges and junctions that represent possible routes from one location to another, and is used to model a linear network.

Domains Specifications for valid values of a field, representing valid value ranges, lists of values, or standard classifications. Domains are used to enforce attribute value integrity.

After the proposed geodatabase design is built and tested, the design is documented using the same visual representation as used in the book of Arctur and Zeiler (2004). An overview of the physical design of the geodatabase is given in Figure 3.2, and a more detailed representation of each element is provided in Appendix B.

The data stored in the geodatabase is organized into two groups: one group contains

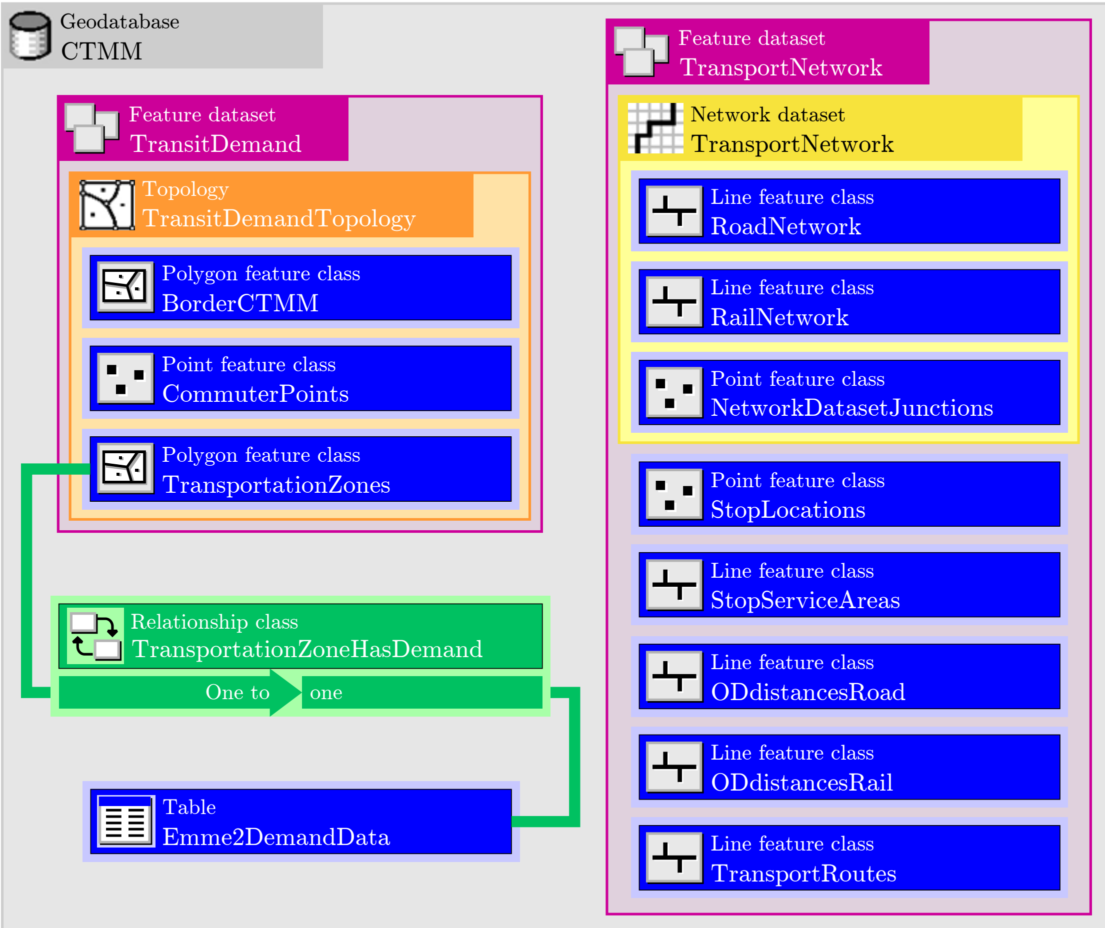


Figure 3.2: Documentation of the physical design of the geodatabase

the data related to the transport network, and the other contains the data related to the demand generation. The feature classes of each group are stored together in a feature dataset. The demand data, which is stored in a table, is linked to the transportation zones through a relationship class. The transit demand feature dataset consists of a topology to ensure that demarcations and commuter points fall into the borders of the study area. The transport network feature dataset contains a network dataset, which enables the GIS to calculate the shortest route from any point in the network to another, while considering the rules of the network, such as one way streets and forbidden turns. The steps followed to generate the OD distances using this network dataset are discussed in the next section.

3.1.3 Generation of the Origin-Destination distances

Once the geodatabase is created, generating the distances between the stops in the network is the main manipulation left to do each time the model is used to optimize a network.

The BSPA provides the GIS with the positions of the stops. The MNDA needs to know the distances between all these stops. The *Network Analyst* tool in *ArcGIS®* provides the option to calculate the distances between all the origin and destination points, using the actual network. By identifying each stop as an origin point and as a destination point, the network analyst procedure generates all the OD distances. Since the transport network consists of a road and a rail network, the procedure is performed twice: once for the road network and once for the rail network. The produced OD distances are stored in the geodatabase, so that they can be used by the MNDA.

Each algorithm in the model requires data from the geodatabase, and the results are stored in the geodatabase. In the following section this interaction is discussed.

3.1.4 Interaction between geodatabase and algorithms

The interactions between the geodatabase and the algorithms are initiated before and after each execution of an algorithm. The first step of the interactions requires the GIS to export the appropriate element of the geodatabase into a shape file format. The *Mapping Toolbox™* in *MATLAB®* enables the algorithms to read files in this format. The results of the algorithms are written to a similar shape file format so that the GIS software can load the data into the geodatabase. The elements in the geodatabase that are used and created by the algorithms are shown in Figure 3.3.

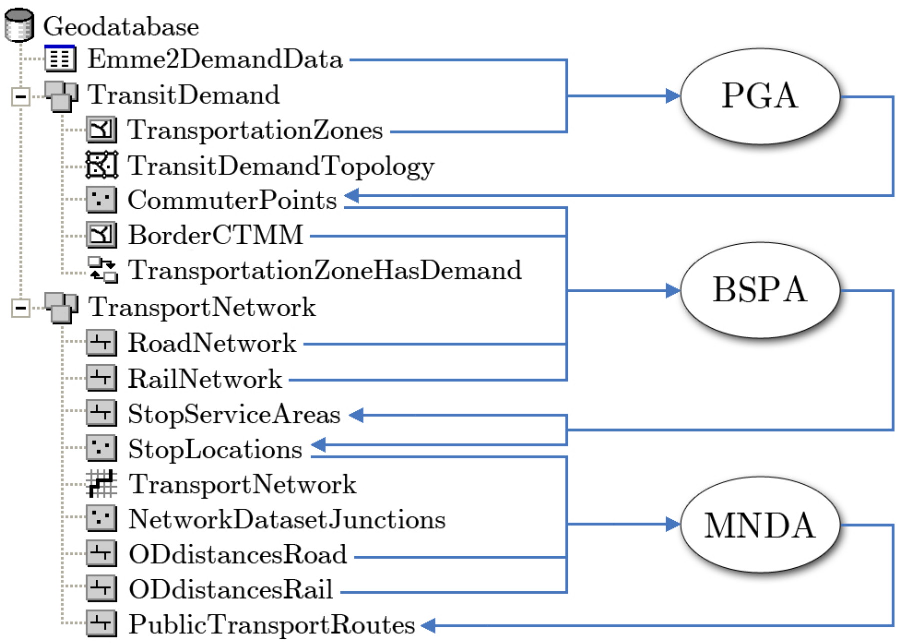


Figure 3.3: Interaction between the algorithms and elements of the geodatabase

3.2 The Population Generation Algorithm

The role of the PGA is to prepare the demand data for the BSPA and MNDA. The BSPA requires that the demand data be represented as point data, in order to calculate the distance from a commuter to its assigned bus stop. The MNDA requires demand data that provides the origin and destination of each commuter, in order to determine the best travel route(s) for each commuter. Therefore, the output of the PGA needs to consist of paired *production* and *attraction* points, with each pair representing one commuter.

The input to the PGA consists of the demand data retrieved from the *EMME/2*® model of the Tshwane Transport Authority, and the division of the CTMM in transportation zones. The structure of the PGA is presented in Figure 3.4. For the purpose of describing the algorithm, the structure is divided into three steps.

1) Transformation of demand matrix

The algorithm starts with loading the input data. The demand data represents the demand per transportation zone during the morning peak hour. However, this data represents the expected average values, which means most values are not nicely-rounded numbers. The aim of the PGA is to produce paired production and attraction points, each representing one 'whole' commuter. Therefore, a transformation of the initial demand data is needed. To achieve this the algorithm follows a couple of steps for each origin transportation zone, in order to assign the demand probabilistically to the destination zones, based on the expected average values from the *EMME/2*® input data.

The algorithm calculates the total demand that originates from the origin transportation zone, and generates a cumulative probability distribution of the demand to all the destination zones. The total demand from the origin zone determines how many commuters will be assigned to the destination zones. For each commuter a random value is generated. The position of the random generated value in the cumulative probability distribution determines the destination zone of that commuter.

2) Creation of production points

For each commuter, new coordinates are randomly generated within the area of a rectangular box touching the outside of the polygon of the origin zone. Since this approach can generate points that lie within this rectangle but not within the borders of the polygon itself, a check is required. If the coordinates are not inside the polygon, new coordinates are generated until a feasible point is found. The points generated in this step represent the production point of a commuter. The coordinates of each production point are stored together with the assigned destination zone number. Once the production points for each origin zone are generated, the attraction points can be generated.

3) Creation of attraction points

For each destination zone, a search is initiated amongst the production points to see how many commuters are assigned to the destination zone under investigation. For each production point found during this search, an attraction point is created with randomly generated coordinates within the borders of the destination zone. The attraction points are stored together with the number of the associated production point. Once this process has been repeated for every destination zone, all the production and attraction points and the list of numbers linking each production/attraction pair are written to file.

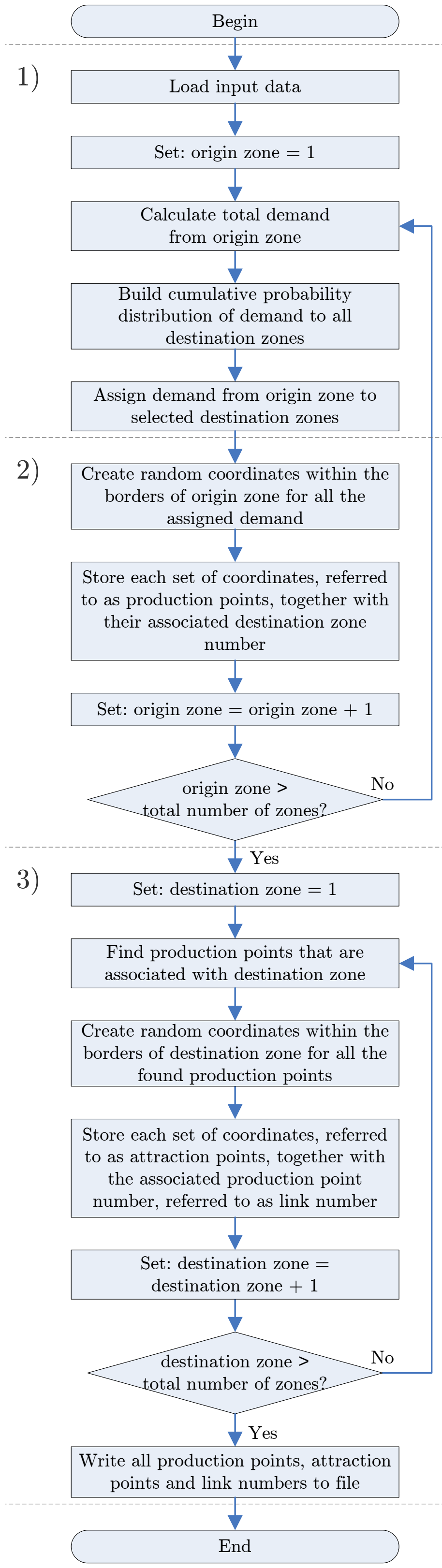


Figure 3.4: Structure of the Population Generation Algorithm

3.3 Conclusion

The development of the GIS and PGA is focused on the approach that uses the demand data from the *EMME/2*® model used by the Tshwane Transport Authority. The GIS component has a central role in storing the geographic data used by the algorithms. To manage this data effectively, a geodatabase design is proposed. The advantages of using a geodatabase are fully utilized by building a topology and network dataset into the design, enabling data integrity and the generation of OD distances respectively. The interactions between the elements of the geodatabase and the three algorithms have been presented, showing which of the elements of the geodatabase are used as input data by the algorithms, and which are created by the algorithms.

The presentation of the PGA component shows how the input demand data is transformed into commuter production and attraction points. For each origin transportation zone, commuters are probabilistically assigned to their destination zones, after which the production and attraction points are generated in the appropriate zones with random generated coordinates.

Chapter 4

The Bus Stop Placement Algorithm

The South African public transport environment requires a unique method for the optimal placement of bus stops. The Bus Stop Placement Algorithm (BSPA) presented in this chapter is based on an idea used for the optimal placement of base stations in a new communication network. The work of Freeman and Minkwitz (2002) is used as a starting point for the development of the algorithm, but changes to the design have been made to convert the model for the use of bus stop placement—such as the variation in the number of bus stops, and the variable service radius for each bus stop.

The algorithm starts with an initial set of bus stops randomly placed within the study area. Bus stops are then moved around, added, or removed as the algorithm searches for a solution that better satisfies the objectives function, while obeying the constraints. The BSPA uses a Simulated Annealing (SA) metaheuristic combined with an intelligent move procedure. The algorithm allows for the integration of existing bus or train lines into the network. Train stations, for example, cannot be easily moved, and when designing a new network the bus stops at these train stations should have a fixed position in the network. Some of the initial bus stops can be classified as fixed; indicating to the algorithm these stops are not to be move or delete.

The chapter starts with the model formulation, where the variables, the objective function, and the constraint of the problem are provided. For the development of the solution approach, the representation of the decision matrix is given and the structure of the algorithm presented. The algorithm is tested on a test area. The appropriate weights of the objective function are retrieved by finding the efficient frontier, and the appropriate input parameter values for the test area are retrieved through a sensitivity analysis.

4.1 Model formulation

The most basic representation of the model consists of the border of the study area, the bus stops, and the commuters. The study area is constrained by a given polygon. Each bus stop, denoted by n, is represented by a coordinate pair with a radius, denoted by (x_n, y_n) and r_n, respectively. Each commuter, q, is represented by its coordinates x_q and y_q. The following notation is used:

Input data and parameters:

$\boldsymbol{B}$	$\triangleq$	Set of coordinates of the polygon representing the border of the study area.
Q	$\triangleq$	Number of commuters in the study area.
(x_q, y_q)	$\triangleq$	Coordinate pair of the q^{th} commuter in the study area, where $q = \{1 \ldots Q\}$.
I	$\triangleq$	Initial number of bus stops.
s_i	$\triangleq$	$\begin{cases} 1 & \text{if initial bus stop } i \text{ has a fixed status, where } i = \{1 \ldots I\}; \\ 0 & \text{otherwise.} \end{cases}$
(x_i, y_i)	$\triangleq$	Coordinate pair of the i^{th} bus stop in the study area, where $i = \{1 \ldots I\}$ (if study area contains fixed bus stops).
$d_{\max}$	$\triangleq$	Maximum walking distance to bus stop.
P	$\triangleq$	Additional penalty factor for unmet demand.
W_1, W_2	$\triangleq$	Weights reflecting the relative importance of commuter distance cost and bus stop placement cost.

Decision variables:

N	$\triangleq$	Number of stops in the current solution.
(x_n, y_n)	$\triangleq$	Coordinate pair of the n^{th} bus stop in the current solution, where $n = \{1 \ldots N\}$.
a_{qn}	$\triangleq$	$\begin{cases} 1 & \text{if commuter } q \text{ is assigned to bus stop } n, \text{ where } q = \{1 \ldots Q\}; \\ 0 & \text{otherwise.} \end{cases}$
d_{qn}	$\triangleq$	Distance between commuter q and bus stop n, where $q = \{1 \ldots Q\}$ and $n = \{1 \ldots N\}$.
r_n	$\triangleq$	Radius of the n^{th} bus stop in the current solution, where $n = \{1 \ldots N\}$.

The problem of placing bus stops in a new network is multi-objective. More bus stops mean shorter walking distances for the commuter; decreasing their overall travel time. On the other hand, more bus stops also mean that the bus will have to stop more often; increasing the overall travel time. An appropriate balance between these two objectives is sought.

The objective function in (4.1) minimizes the sum of the total walking distance and the number of bus stops in the network.

$$\min z = \quad W_1 \sum_{q=1}^{Q} \left[\sum_{n=1}^{N} (a_{qn} d_{qn}) + \left(1 - \sum_{n=1}^{N} a_{qn} \right) d_{\max} P \right]^2 + W_2 N \tag{4.1}$$

The first component represents the objective to decrease the walking distance for commuters to a bus stop. A certain cost is created by taking the sum of the square of the walking distance for every commuter. Taking the square of the walking distance ensures that large walking distances are penalized more heavily than shorter walking distances. This is to simulate commuters' behaviour better, since long walking distances are experienced as a much heavier burden than shorter distances. Each commuters that is not assigned to any bus stop generates a predetermined value for the 'walking distance' in the objective function, calculated by multiplying the maximum walking distance, $d_{\max}$, by the additional penalty factor for unmet demand, P. The second component represents the objective to reduce the number of bus stops in the network. Both the components are weighted to adjust their relative importance.

The model formulation of the BSPA includes two main constraints. Both of them are discussed below.

The Maximum Walking Distance Constraint in (4.2) ensures that no commuter is connected to a bus stop that is further away than the maximum walking distance, d_{max}. For a public transport network to be a viable option for travelling from A to B, the distance to the first connection point of that network needs to be within a certain range. Therefore, a commuter can only be connected to a certain bus stop if the distance between them does not exceed a certain value. The constraint in (4.3) ensures that a_{qn} is a binary value.

$$a_{qn} d_{qn} \leq d_{\max} \qquad \forall q = \{1 \dots Q\}, n = \{1 \dots N\} \tag{4.2}$$

$$a_{qn} \in \{0, 1\} \qquad \forall q = \{1 \dots Q\}, n = \{1 \dots N\} \tag{4.3}$$

The Assignment Constraint in (4.4) restricts commuters from being assigned to multiple bus stops. If a commuter is surrounded by multiple bus stop within the maximum walking distance, the commuter is assigned to the closest, resulting in $\sum_{n=1}^{N} a_{qn} = 1$. If no bus stop lays within the maximum walking distance of the commuter, the commuter is classified as unmet demand, resulting in $\sum_{n=1}^{N} a_{qn} = 0$.

$$\sum_{n=1}^{N} a_{qn} \leq 1 \qquad \forall q = \{1 \dots Q\} \tag{4.4}$$

4.2 Solution approach

The solution approach chosen for the BSPA consists of an SA algorithm, combined with an intelligent procedure to move through the solution space. Before the main components of the algorithm are discussed, the representation of the decision matrix used in *MATLAB*® is given. The decision matrix presented below is generated by the algorithm to keep track of the details of the current solution.

$$A = \begin{bmatrix} x_1 & y_1 & r_1 & s_1 \\ x_2 & y_2 & r_2 & s_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & r_N & s_N \end{bmatrix}$$

where:

$N \triangleq$ Number of stops in the current solution.

$x_n \triangleq$ The x-coordinate of bus stop n, where $n = \{1 \dots N\}$.

$y_n \triangleq$ The y-coordinate of bus stop n, where $n = \{1 \dots N\}$.

$r_n \triangleq$ The radius of the service area of bus stop n in meters, where $n = \{1 \dots N\}$.

$$s_n \triangleq \begin{cases} 1 & \text{if bus stop } n \text{ has a fixed status, where } n = \{1 \dots N\}; \\ 0 & \text{otherwise.} \end{cases}$$

The structure of the BSPA is shown in Figure 4.1. For the purpose of describing the algorithm, the structure is divided into six steps.

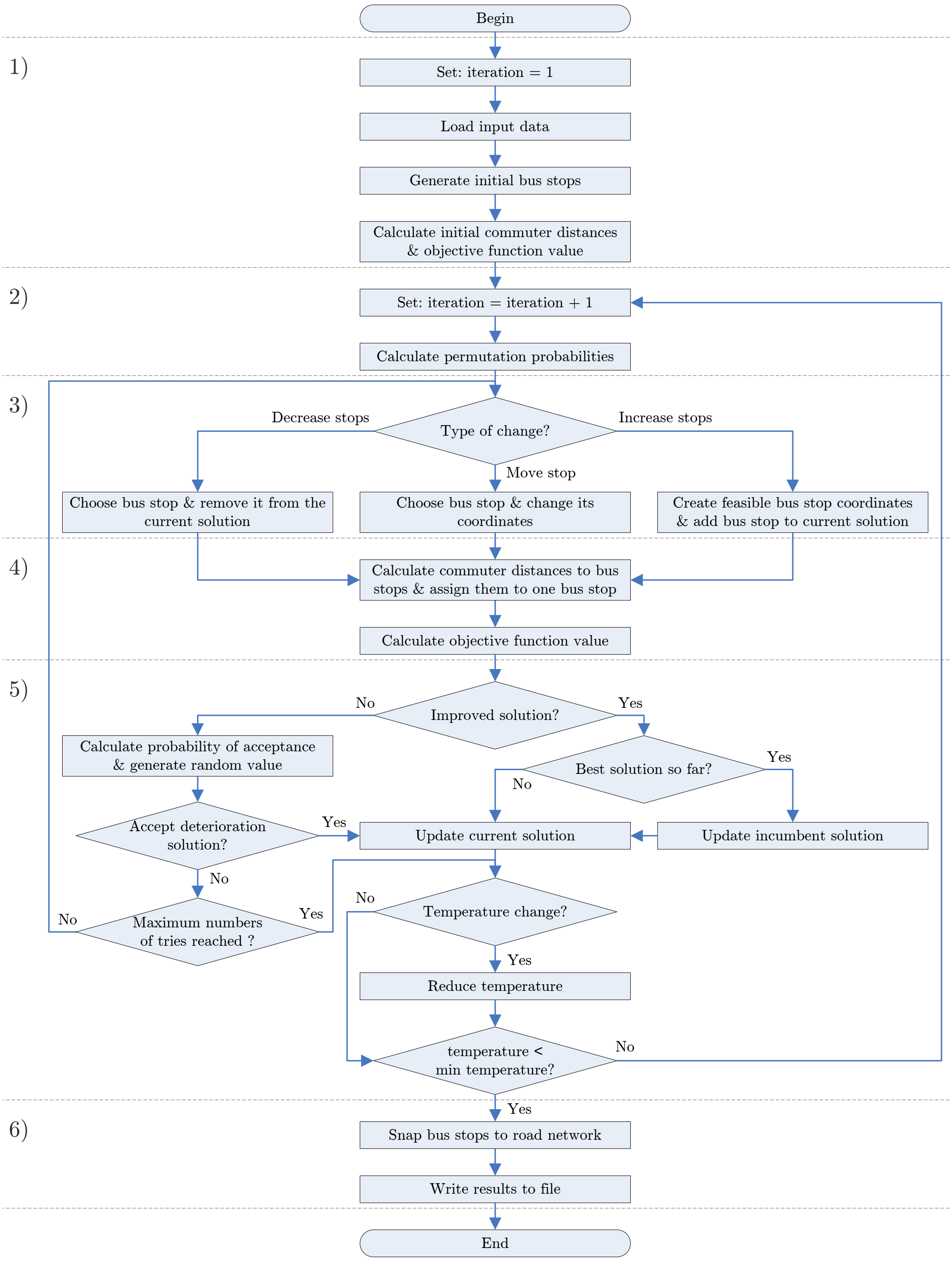


Figure 4.1: The Bus Stop Placement Algorithm

1) Initialization of the algorithm

In the initialization phase the algorithm begins with loading the input data: geographic data indicating the border of the study area as well as the commuter coordinates. The initial bus stops are generated with random x- and y-coordinates. If the study area consists of any bus stops that need to be fixed, the coordinates of these bus stops need to be provided manually. The algorithm then calculates the commuter distances of the initial solution, assigns the commuters to a bus stop, and calculates the initial objective function value.

2) Adjustment of perturbations weights

From this step on, the iterative search for a better solution starts. After a predetermined number of iterations, the algorithm (re)calculates the perturbation weights. The weights represent the probability that the algorithm will choose one of the following perturbation: 1) decrease the number of bus stops; 2) change the coordinates of a bus stop; or 3) increase the number of bus stops. The weights are calculated based on the effect of the perturbations on the objective function value. A perturbation resulting in a good solution is assigned a higher weight than perturbations resulting in bad solutions. The recalculation of the weights after a predetermined number of iterations provides the algorithm with the ability to initiate a more intelligent choice of change while searching for a better solution, as it enables the algorithm to react on its current position in the solution space.

The effect of each perturbation on the objective function value provides the perturbations a first, second, or third place. The first place produces a perturbation weight of 4, second place a weight of 2, and third place a weight of 1. However, the perturbation weight of the MOVE STOP type of change is multiplied by two, since there is only one type of perturbation that changes the coordinates of the bus stops, while there are two types of perturbations that change the number of bus stops. The cumulative perturbation probabilities are then calculated according to the perturbation weights. If, for example, the predetermined number of iterations is set to 10, the perturbation probabilities are used for 10 iterations by the procedure discussed in the next step, until new perturbation weights are calculated.

3) Selecting the type of change

In this step, one of the three types of changes is chosen, probabilistically based on the perturbation weights. The three types of changes are discussed below.

Move stop A random bus stop in the current solution is selected. The x- and y-coordinates are changed in a random direction by a random distance. The new coordinates are iteratively generated until they fall within the border of the study area.

Decrease stops A randomly selected bus stop is removed from the current solution.

Increase stops A new bus stop with randomly generated coordinates is added to the current solution. Its coordinates are iteratively generated until they fall within the border of the study area.

4) Commuter distance and objective function calculation

In this step, the effects of the change are implemented and calculated. The distance from every commuter to every bus stop d_{qn} is determined and every commuter is subsequently assigned to the closest bus stop. If commuters cannot be assigned to any bus stop because the distance to the closest bus stop exceeds the maximum walking distance, this commuter is categorized as *unmet demand*. The objective function value is calculated by using (4.1) on page 29.

5) Accepting or rejecting proposed change

The objective function value of the new solution is compared with the previous solution. If an improvement is made, the new solution becomes the current solution. The algorithm then also checks if this improved solution is the best one so far. If so, the incumbent solution is updated. If the new solution is not an improvement, the algorithm has a probability of still accepting the change. This probability is determined by the temperature of the cooling schedule and the amount of deterioration, as explained in Section 2.1.1 on the SA metaheuristic. If the change is rejected, the algorithm starts over and proposes another change until a move is accepted or the maximum number of tries is reached. If the maximum number of tries is reached and still no change is accepted, the previous solution is kept. After a predetermined number of iterations the temperature is reduced.

This is in order to decrease the probability of accepting deteriorating moves as more and more iterations are done, enabling the algorithm to converge to a local and maybe global optimal solution. Once the temperature has reached the minimum allowed temperature the search for a better solution is finished.

6) Snapping the bus stops to actual roads

The last step of the algorithm consists of adjusting the positions of the bus stops to realistic positions. This is done by comparing the positions of the bus stops to a road network of the area, obtained from the Geographic Information System (GIS) geodatabase. For each bus stop the closest road is found and the position changed. Most road networks classify their roads according to their size, importance, and traffic flow. This information can be used to specify the road type to which the bus stops should be aligned. For example, highways and dirt roads can be excluded from possible roads to snap to. Once this is done the final solution is written to file.

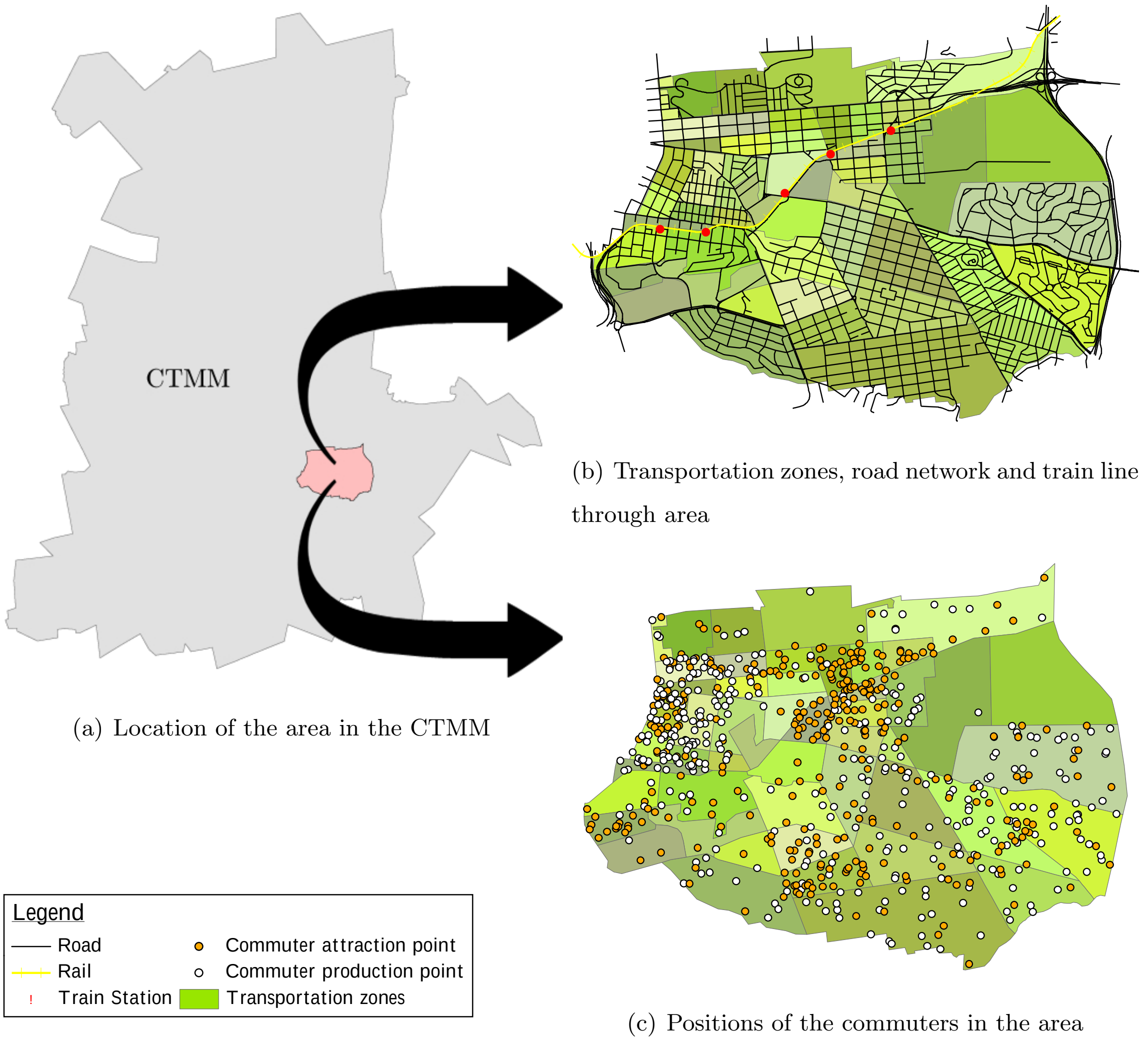
4.3 Algorithm performance

To evaluate the performance of the algorithm and to find the appropriate values for the weights of the two conflicting objectives of the objective function, the BSPA is tested on a small part of the City of Tshwane Metropolitan Municipality (CTMM), as indicated in Figure 4.2. The effects of the weights of the two conflicting objectives are analysed by looking at the efficient frontier of this multi-objective problem. The appropriate values for the weights are chosen and used in the sensitivity analysis, where the effect of the parameters guiding the algorithm is analysed. Finally the network proposed by the BSPA is presented.

4.3.1 Outline of test area

The selected test area consists of nearly 40 km^2 of land. The west side of the area borders the Central Business District (CBD) of the CTMM. The area accommodates mostly residential areas, some recreation areas, and a rail line. The location of the test area in the CTMM is shown in Figure 4.2(a).

The outline of the test area, as stored in the geodatabase, is shown in Figure 4.2(b). The different green coloured areas represent the transportation zones, and the thin black



(a) Location of the area in the CTMM

(b) Transportation zones, road network and train line through area

(c) Positions of the commuters in the area

Figure 4.2: Outline of the test area for the BSPA

lines the road network of the area. The thicker yellow line represents the rail line through the area, and the round red points the train stations.

The demand data is retrieved from the transport authority of the CTMM and manipulated into point data by the Population Generation Algorithm (PGA). The area consists of 690 commuter points: 345 production and 345 attraction points. For the purpose of testing the algorithm, only the production and attraction demand within the test area is considered, which represents around 5% of the total transit demand in the area. Figure 4.2(c) shows the positions of the commuter points in the test area.

The maximum walking distance to a bus stop, d_{max}, is set to 2000m. In most literature this value is much lower. Saka (2001), for example, suggests a maximum walking distance of 800m, and in the work of Alterkawi (2006) the walking distance is limited to between 300m and 400m. However, due to the disparate South African commuting circumstances and poorly developed public transport services, a higher value for d_{max} will produce a network for this test area that is similar to current networks. The additional penalty factor for unmet demand, P, is set to 1.2.

The values for the weights W_1 and W_2 reflecting the relative importance of commuter distance cost and bus stop placement cost are chosen after the analysis of the efficient frontier, which is discussed in the next section.

4.3.2 Efficient frontier

The BSPA is a multi-objective metaheuristic that tries to optimize a problem with two conflicting objectives: reduction of the distance between commuters and bus stops; and reduction of the number of bus stops. For multi-objective problems there is no single best solution, but a whole range of 'best' solutions, called *efficient points*, depending on the weights given to the different objectives. A solution is defined as an efficient point if none of the objectives can be further improved, without deterioriting any of the other objectives. The *efficient frontier* is defined as the collection of efficient points for a multi-objective optimization model (Rardin, 1998).

In this section the efficient frontier of the BSPA is determined. The different solutions on the efficient frontier are analysed in order to determine the most appropriate values for the two objective weights W_1 and W_2. Although the efficient frontier is specific to this test area, the values retrieved for the objective weights can also be used for the optimization of the whole CTMM area, as long as the precentage of the total demand is kept the same.

The efficient frontier is retrieved by removing the objective that reduces the number of bus stops from the objective function and replacing it by a constraint that limits the number of bus stops in the network. The algorithm is then run with different values for this constraint. This translates into the algorithm trying to find the best solution (the lowest commuter distance cost) for each maximum number of allowed bus stops. This number ranges from 5 to 690, with 5 being the lowest number since the area consists of five train stations with a fixed status and 690 being the largest number since 690 commuter points are located in the area, providing each commuter point with its own bus stop. For each value of the constraint three runs are done, and the best result is shown in Figure 4.3.

We will now look at certain points on the efficient frontier and determine which one is the most appropriate. For illustrative reasons, the points corresponding with 10, 19, 30, 40, and 50 maximum allowed bus stops are examined and shown in Figure 4.4.

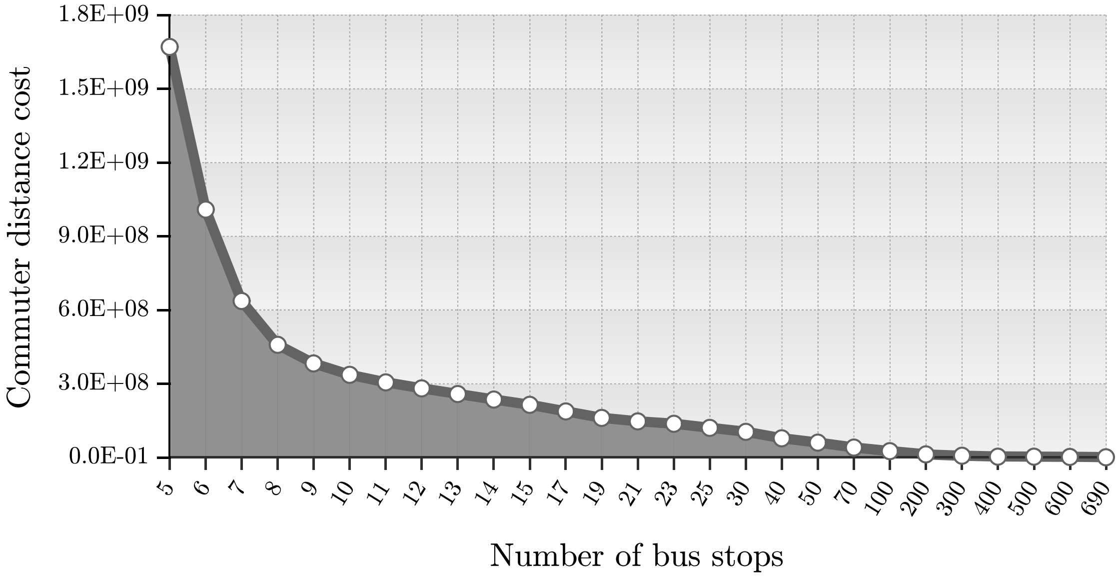


Figure 4.3: Efficient frontier of multi-objective metaheuristic on test area

The efficient point chosen as the most appropriate one depends on the requirements of the network that needs to be optimized. For the purpose of this project the efficient point associated with the constraint of maximum 30 bus stops is chosen. The objective function value at this efficient point amounts to 103 891 127. The solution consists of 30 bus stops, which means that each bus stop should be assigned a cost of 103 891 127/30 = 3 463 038 in order to create the same kind of solution. For the sensitivity analysis the value for the weight associated with the cost for bus stops, W_2, is rounded to 3.5 million and the value for W_1 is set to 1.

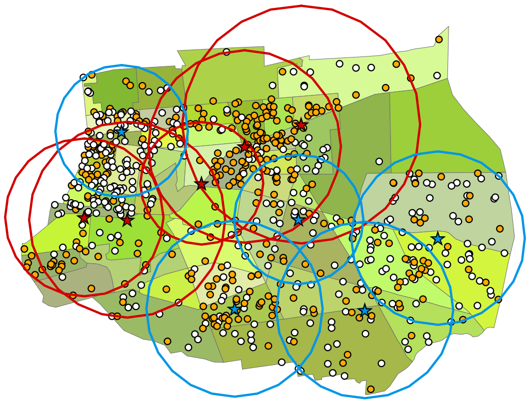
4.3.3 Sensitivity analysis

The sensitivity analysis is performed on seven parameters that guide the algorithm, of which the first four are part of the SA cooling schedule. The role of each parameter in the algorithm is discussed below.

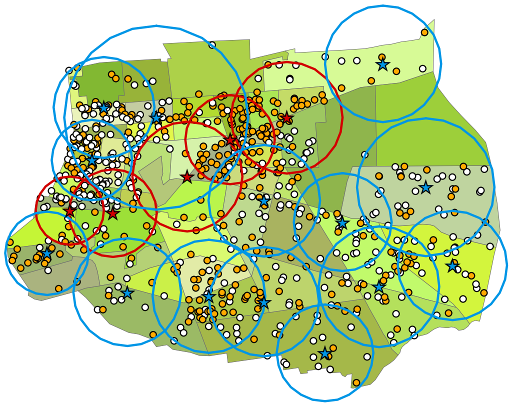
tempReductionFreq Determines after how many iterations the temperature is reduced.

tempReductionFactor Determines by how much the value of the temperature is reduced.

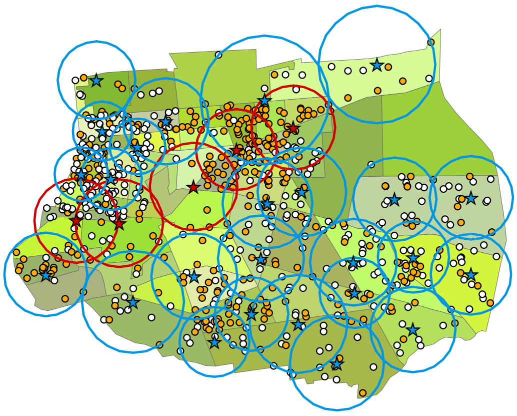
tempObjectiveFactor Used to calculate the initial temperature, by multiplying this



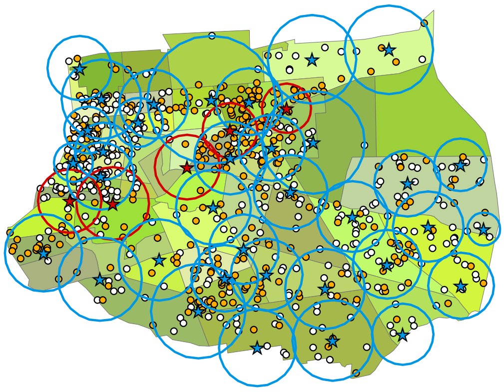
(a) Constraint of maximum 10 bus stops; resulting in a average walking distance of 618m



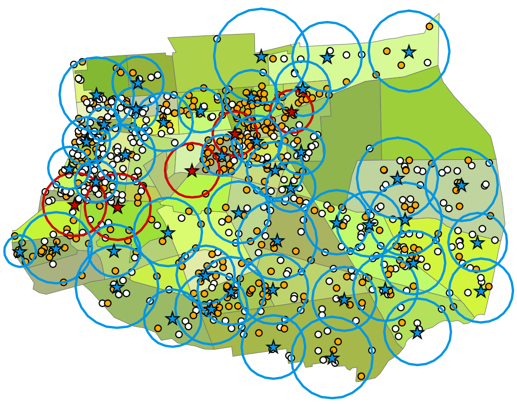
(b) Constraint of maximum 19 bus stops; resulting in a average walking distance of 434m



(c) Constraint of maximum 30 bus stops; resulting in a average walking distance of 353m



(d) Constraint of maximum 40 bus stops; resulting in a average walking distance of 299m



(e) Constraint of maximum 50 bus stops; resulting in a average walking distance of 264m

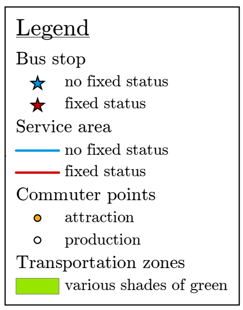


Figure 4.4: The resulting bus stop positions for five points on the efficient frontier. For a better comparison the bus stops are not aligned with the road network.

parameter by the initial objective function value.

minTemp Used as the stopping criterion of the algorithm. If the temperature reaches the value of *minTemp* the search for a better solution is complete.

initialNrStops Determines the initial number of bus stops placed in the area.

maxPossibleMoves Whenever a change to the current solution is proposed and not accepted, the algorithm proposes another change. This cycle is repeated until a change is accepted or the number of tries reaches the value of *maxPossibleMoves*.

perburbationFreq Determines after how many iterations the perturbation weights are recalculated.

For each parameter five values are chosen that are tested during the sensitivity analysis (see Table 4.2). Since seven different parameters are used, this results in $7^5 = 16\ 807$ possible parameter settings. To test all these parameter combinations is a very time-consuming activity and is beyond the scope of this project. Therefore, an approach is chosen that first determines a reasonably good and stable combination of parameter values, and then uses this combination as the baseline to evaluate the effect of the parameters values on the objective function value.

Table 4.2: Parameter values chosen for the sensitivity analysis of the BSPA

Parameter	Values	Parameter	Values
tempReductionFreq	5, 15, 25, 35, 45	initialNrStops	10, 20, 30, 40, 50
tempReductionFactor	0.75, 0.8, 0.85, 0.9, 0.95	maxPossibleMoves	5, 10, 20, 50, 100
tempObjectiveFactor	0.05, 0.1, 0.5, 1, 5	perburbationFreq	1, 5, 10, 50, 100
minTemp	1, 5, 10, 50, 100		

To find the baseline parameter setting, 50 combinations are randomly chosen and run three times. The average result of the three runs is compared, as well as the Mean Absolute Deviation (MAD)—a measurement of the repeatability of the solution. The MAD is calculated according to (4.5).

$$\text{MAD} = \frac{1}{R}\sum_{i=1}^{R}|x_i - \bar{x}| \tag{4.5}$$

where:

R	$\triangleq$	The number of runs done with one parameter setting.
x_i	$\triangleq$	The i^{th} solution of runs done with one parameter setting, where $i = \{1 \ldots R\}$.
$\bar{x}$	$\triangleq$	The average of all the runs done with one parameter setting.

The results of running the algorithm with 50 different parameter settings are shown in Figure 4.5. The MAD for each parameter setting is given in the percentage of the objective function value. The values used for each parameter setting are provided in Appendix C. The 30^{th} parameter setting generates the best results and also the best MAD %, and is therefore used as the baseline for the rest of the sensitivity analysis. The values corresponding to this parameter setting are shown in Table 4.3.

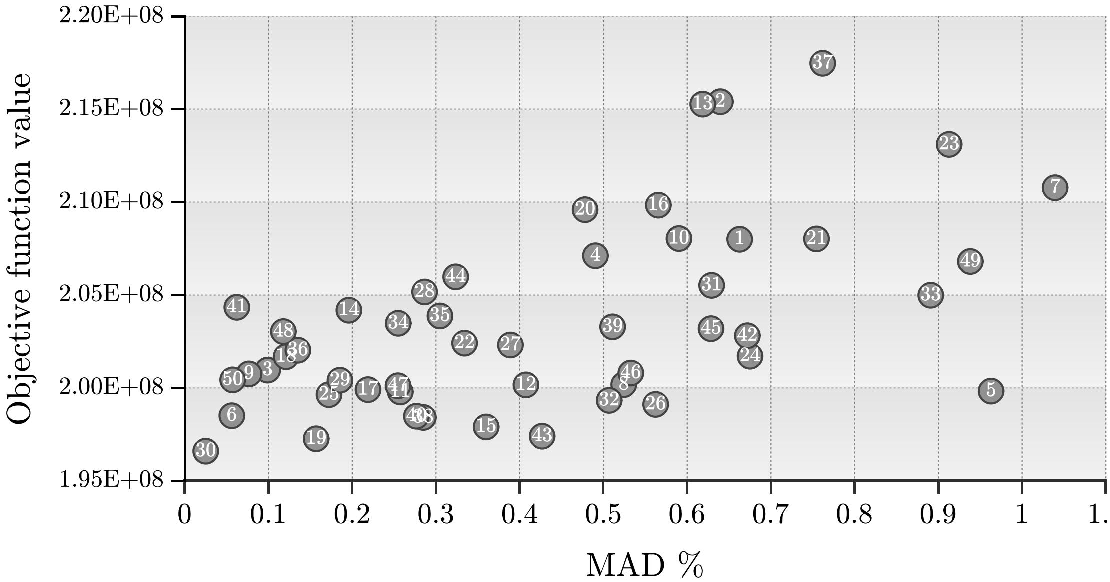


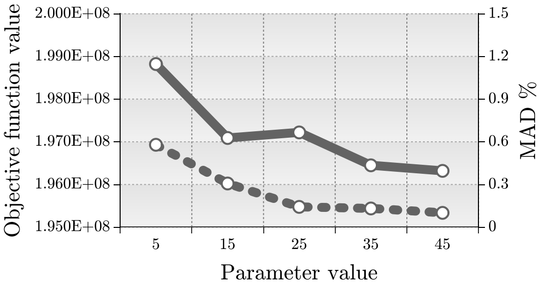
Figure 4.5: Results of 50 different parameter settings. Each point on the graph represents the average objective function value and MAD % retrieved after three runs of one parameter setting. The number on each point in the graph refers to the ID-number given to each parameter setting, and can be used to find the corresponding parameter values of each point through Appendix C.

Table 4.3: Values used as the baseline parameter setting for the sensitivity analysis

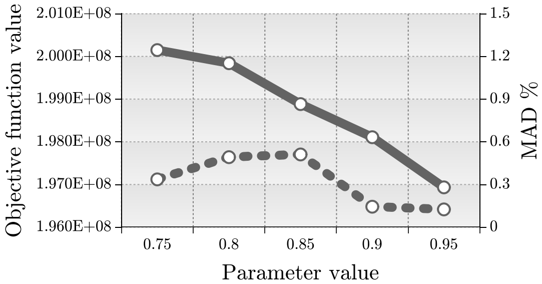
Parameter	**Value**	**Parameter**	**Value**
tempReductionFreq	15	initialNrStops	20
tempReductionFactor	0.95	maxPossibleMoves	100
tempObjectiveFactor	0.05	perburbationFreq	100
minTemp	5		

Each time during the sensitivity analysis five runs are executed, while changing the value of one parameter and keeping all the values of the other parameters constant by assigning them the baseline parameter setting as shown in Table 4.3. The results of the sensitivity analysis are presented in Figure 4.6.

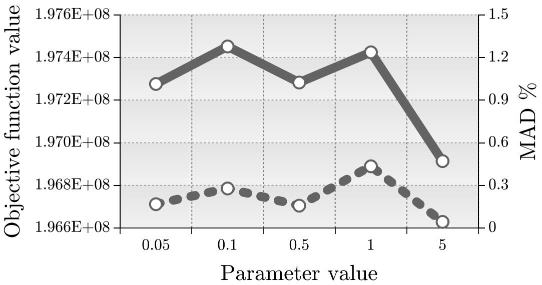
The *tempReductionFreq*, *tempReductionFactor* and *tempObjectiveFactor* parameters, shown in Figures 4.6(a)–4.6(c), all generate the best results with parameter values that result in the highest number of iterations—45, 0.95, and 5 respectively. This can be expected since it provides the algorithm with more opportunities to find a better solution. Increasing these parameter values further might enable the algorithm to find even better solutions, but also increases the computational resources needed to perform a run. The *minTemp* parameter in Figure 4.6(d) also shows a trend towards parameter values resulting in the highest number of iterations, except that an optimum is found at a value of 5 instead of 1, indicating that a further decrease of the minimum temperature of the algorithm is unnecessary. For the *intialNrStops* parameter in Figure 4.6(e) no clear pattern is discovered. The optimum solution and lowest MAD % is found at a value of 40. In Figure 4.6(f) the *maxPossibleMoves* parameter shows a trend towards better solutions with the increase of its value. Higher values provide the algorithm with more opportunities within one iteration to try to find a change to the current solution that will be accepted. The value of 100 also provides a good MAD %. For the *perturbationFreq* parameter in Figure 4.6(g) the best solutions are generated with a value of 50, indicating that occasional adjustment of the perturbation weights generates better results than a continuous adjustment, something that was not initially anticipated. The parameter values found through the sensitivity analysis are used to generate a near-optimal solution for the test area in the next section.



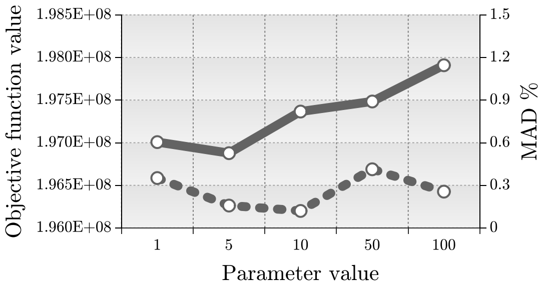
(a) tempReductionFreq



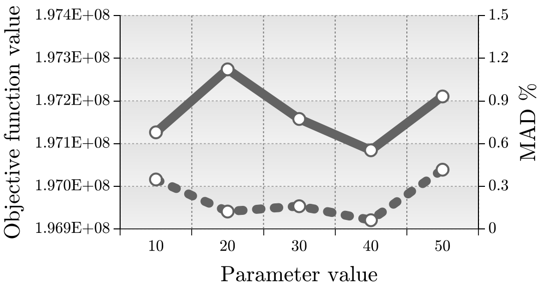
(b) tempReductionFactor



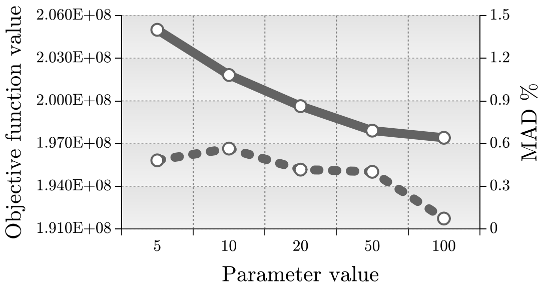
(c) tempObjectiveFactor



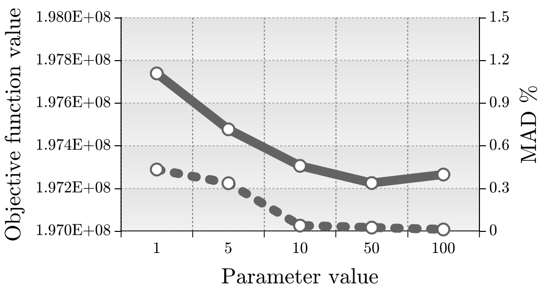
(d) minTemp



(e) initialNrStops



(f) maxPossibleMoves



(g) perturbationFreq

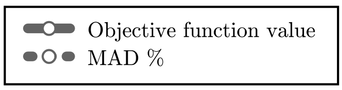


Figure 4.6: Sensitivity analysis on seven parameters of the BSPA

4.3.4 Proposed solution

The proposed solution is generated by running the algorithm ten times with the parameter values shown in Table 4.4. The best solution produced during these ten runs is presented in Figure 4.7. The solution consists of 30 bus stops and an average walking distance of 327m.

Table 4.4: Parameter values retrieved through the sensitivity analysis

Parameter	Value	Parameter	Value
tempReductionFreq	45	initialNrStops	40
tempReductionFactor	0.95	maxPossibleMoves	100
tempObjectiveFactor	5	perburbationFreq	50
minTemp	5		

This particular run consists of 17 145 iterations and produces a final solution with an objective function value of 195 678 906. The total run took 1 hour and 25 minutes on one node of the *Velocity* cluster (3 GHz processor). The objective function value of the current solution and the incumbent solution during each iteration are shown in

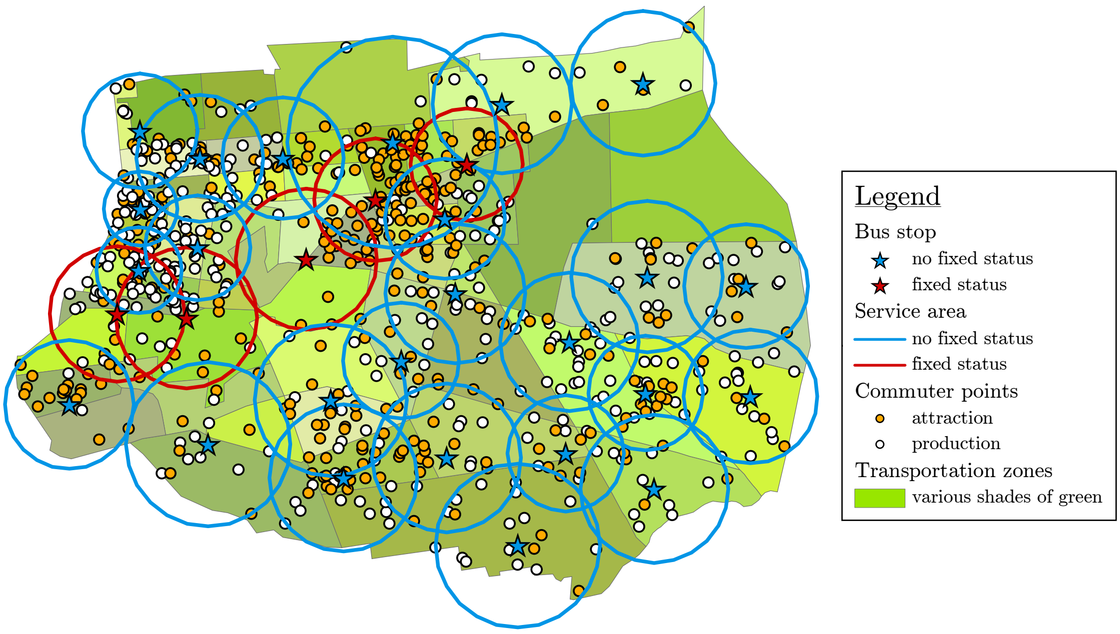


Figure 4.7: Best solution retrieved after ten runs with optimized parameter setting. Bus stops are aligned with road network.

Chapter 5

The Multimodal Network Design Algorithm

In order to model a public transport system as accurately as possible, it is important to include all the modes of public transport into the model—especially in the case of South Africa, where all the different modes of public transport currently compete for the same market share. For the development of the Multimodal Network Design Algorithm (MNDA) the following changes are made to the work of Fan and Machemehl (2006b):

Multimodal network The algorithm considers various modes of public transport, such as double- and single-decker buses, minibus-taxis, and trains. Each mode in the algorithm is described by the characteristics of its vehicles, which include their specific seating capacity, load factor, average travel speed, etc. Rail lines and fixed bus routes are included by enabling the algorithm to classify these routes as fixed, prohibiting the algorithm from removing them from the network.

Shortest travel time The algorithm provides the commuter with the choice of making a transfer instead of always choosing the path with the fewest transfers. Paths with the fewest transfers no longer automatically receive all the demand. The algorithm searches for the fastest travel path, while considering all the possible paths, and then assigns the demand to the quickes routes.

Asymmetric demand The algorithm considers asymmetric demand instead of only symmetric demand. It is more realistic to assume that in one hour the demand production and attraction of a certain point in the network are not exactly equal.

This chapter starts by presenting the model formulation; defining the variables, objective function and the constraints of the model. In the solution approach each of the three main procedures of the MNDA are discussed. The algorithm is then tested on a test network: the outline of the test network is provided, the results of the sensitivity analysis are given, and the proposed network is presented.

5.1 Model Formulation

The basic representation of the public transport system as formulated for the MNDA consists of stops, links, and routes. Stops are points in the network through which commuters gain access to the public transport network. Note that stops in this chapter are no longer called 'bus stops', since these stops can provide access to other public transport modes than just buses—for example, trains. Links are connectors between stops that contain the distance between two stops. Routes are a collection of stops that are linked in a specific order. The model designs the network on a per-hour basis. The distances between the stops are converted to travel time by considering the average speed of the vehicles traversing the routes. The following notations are used:

Input data and parameters:

N	$\triangleq$	Number of stops in the network.
d_{ij}	$\triangleq$	Travel demand between stops i and j, where $i, j = \{1 \dots N\}$.
l_{ij}	$\triangleq$	Distance between stops i and j, where $i, j = \{1 \dots N\}$.
K	$\triangleq$	Number of vehicle types.
S_k	$\triangleq$	Seating capacity of vehicle type k, where $k = \{1 \dots K\}$.
$F_k^{\max}$	$\triangleq$	Maximum load factor for vehicle type k on any route, where $k = \{1 \dots K\}$.
V_k	$\triangleq$	Average speed at which type k vehicles travel, where $k = \{1 \dots K\}$.
O_k	$\triangleq$	Per hour operating cost of type k vehicles, where $k = \{1 \dots K\}$.
$L_{\max}$	$\triangleq$	Maximum length of any route in the network.
$L_{\min}$	$\triangleq$	Minimum length of any route in the network.
$l_{\max}$	$\triangleq$	Maximum distance between two directly connected stops.
T	$\triangleq$	Transfer penalty.
C	$\triangleq$	Value of the commuters' time spend travelling (per hour).
U	$\triangleq$	Cost for unsatisfied demand.

$W_1, W_2, W_3 \triangleq$ Weights reflecting the relative importance of user cost, operator cost and unsatisfied demand cost respectively.

$M \triangleq$ Number of routes in the network.

Decision variables:

$L_m \triangleq$ Length of route m, where $m = \{1 \ldots M\}$.

$\boldsymbol{P_{ij}} \triangleq$ Set of paths used to serve the demand from stop i to stop j, where $i, j = \{1 \ldots N\}$.

$d_{ij}^p \triangleq$ Demand from stop i to stop j allocated to path p, where $i, j = \{1 \ldots N\}$ and $p \in \boldsymbol{P_{ij}}$.

$t_{ij}^p \triangleq$ Travel time between stops i and j over path p, where $i, j = \{1 \ldots N\}$ and $p \in \boldsymbol{P_{ij}}$.

$\boldsymbol{\Omega_m} \triangleq$ Set of links assigned to route m, where $m = \{1 \ldots M\}$.

$D_m^l \triangleq$ Demand allocated to link l on route m, where $m = \{1 \ldots M\}$ and $l \in \boldsymbol{\Omega_m}$.

$Q_m^{\max} \triangleq$ Maximum flow on the critical link of route m, where $m = \{1 \ldots M\}$.

$A_m \triangleq$ Number of vehicles on route m, where $m = \{1 \ldots M\}$.

$f_m \triangleq$ Frequency of vehicles on route m, where $m = \{1 \ldots M\}$.

$$o_m^k \triangleq \begin{cases} 1 & \text{if vehicle type } k \text{ is operating on route } m, \text{ where } k = \{1 \ldots K\} \text{ and } m = \{1 \ldots M\}; \\ 0 & \text{otherwise.} \end{cases}$$

$$c_{ij} \triangleq \begin{cases} 1 & \text{if stop } i \text{ is connected with stop } j, \text{ where } i, j = \{1 \ldots N\}; \\ 0 & \text{otherwise.} \end{cases}$$

The key aspects of the model formulation are discussed below. The aim of the objective function presented in (5.1) is to minimize the total cost.

$$\begin{aligned} \min z = \quad & W_1 \left(C \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{p \in \boldsymbol{P_{ij}}} d_{ij}^p t_{ij}^p \right) + W_2 \left(\sum_{m=1}^{M} \sum_{k=1}^{K} O_k A_m o_m^k \right) \\ & + W_3 \left(U \sum_{i=1}^{N} \sum_{j=1}^{N} \left(d_{ij} - \sum_{p \in \boldsymbol{P_{ij}}} d_{ij}^p \right) \right) \end{aligned} \tag{5.1}$$

The first part of the objective function represents the *total user cost*, which is calculated by multiplying the demand from stop i to stop j over path p with the total travel time over that path. The second part represents the *total operator cost*, which consists of the cost of a vehicle per hour multiplied by the number of vehicles on that route. The third part represents the *unsatisfied demand*, which is retrieved by subtracting the demand allocated to the various paths from the total demand.

The four main constraints included into the model formulation of the MNDA, and their related sub-constraints, are discussed below.

The FREQUENCY CONSTRAINT shown in (5.2) ensures that the frequencies of the vehicles operating on the routes is adequate for handling the demand allocated to the routes. The frequency depends on the demand allocated to the critical link, as calculated by (5.3), and the type of vehicle operating on the route. A minimum frequency of one vehicle per hour applies for each route as indicated by (5.4). Each route is assigned one vehicle type as indicated by (5.5). The constraint in (5.6) ensures that o_m^k is a binary value.

$$f_m = \sum_{k=1}^{K} \frac{Q_m^{\max}}{S_k F_k^{\max}} o_m^k \qquad \forall m = \{1 \dots M\} \tag{5.2}$$

$$Q_m^{\max} = \max_{l \in \boldsymbol{\Omega_m}} \left\{ D_m^l \right\} \qquad \forall m = \{1 \dots M\} \tag{5.3}$$

$$f_m \geq 1 \qquad \forall m = \{1 \dots M\} \tag{5.4}$$

$$\sum_{k=1}^{K} o_m^k = 1 \qquad \forall m = \{1 \dots M\} \tag{5.5}$$

$$o_m^k \in \{0, 1\} \qquad \forall m = \{1 \dots M\}, k = \{1 \dots K\} \tag{5.6}$$

The VEHICLE CONSTRAINT shown in (5.7) ensures that sufficient vehicles are assigned to each route based on the vehicle frequency, f_m, and round-trip-time, $\frac{2L_m}{V_k}$, on each route. The number of vehicles on each route is restricted to whole numbers and a minimum number of one by (5.8).

$$A_m \geq \sum_{k=1}^{K} f_m \frac{2L_m}{V_k} o_m^k \qquad \forall m = \{1 \dots M\} \tag{5.7}$$

$$A_m \in \mathbb{N} \qquad \forall m = \{1 \dots M\} \tag{5.8}$$

The ROUTE LENGTH CONSTRAINT prevents the algorithm from considering routes that are too short or too long. This constraint is important from both the commuters' and operators' perceptive. Commuters do not favour routes that are too long, as this will increase the probability of vehicles running late. On the other hand, operators do not favour routes that are too short, as this threatens the efficiency of the network. The constraint formulated in (5.9) ensures that route length is kept within the user-defined range.

$$L_{\min} \leq L_m \leq L_{\max} \qquad \forall m = \{1 \ldots M\} \tag{5.9}$$

The Connection Constraint ensures that if no input is provided to the algorithm about which stops are directly connected, the algorithm doesn't classify stops too far away from one another as *directly connected stops*. The constraint is formulated in (5.10). The constraint in (5.11) ensures that c_{ij} is a binary value.

$$\sum_{i=1}^{N} \sum_{j=1}^{N} l_{ij} c_{ij} \leq l_{\max} \tag{5.10}$$

$$c_{ij} \in \{0, 1\} \qquad \forall i, j = \{1 \ldots N\} \tag{5.11}$$

5.2 Solution approach

The solution approach chosen for the MNDA is based on the work of Fan and Machemehl (2006b). The algorithm consists of the following procedures: 1) the Initial Candidate Route Set Generation Procedure (ICRSGP); 2) the Network Analysis Procedure (NAP); and 3) the Simulated Annealing Search Procedure (SASP). Figure 5.1 shows the outline and data flow of the MNDA.

Figure 5.2 shows the representation of the decision matrix used by the algorithm. The rows represent the routes, m, of the current solution. The columns, representing the characteristics of the routes, are discussed below.

Vehicle type Refers to the type of vehicle selected to operate on route m.

Vehicle frequency Refers to the number of vehicles operating per hour on route m.

Route number Indicates the position of the route m in the solution space.

Route access Indicates the type of access route m provides for the current Origin-Destination (OD) demand. This can be direct, origin, destination or other (possible transfer) access.

Route stops Is the list of all the stops that make up route m.

Demand on links in same direction Indicates the amount of transit demand traversing the links on route m in the *same* direction as the sequence of the stops.

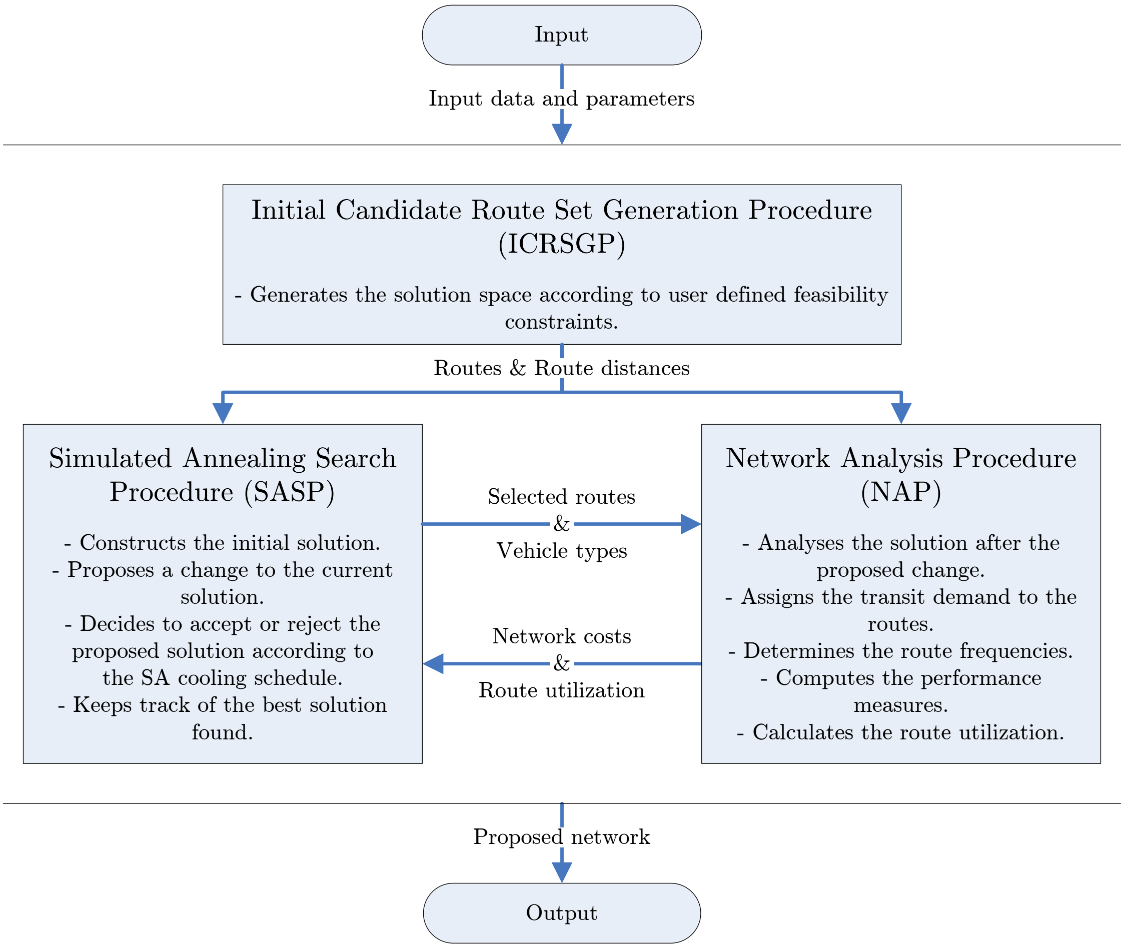


Figure 5.1: Overview of the Multimodal Network Design Algorithm

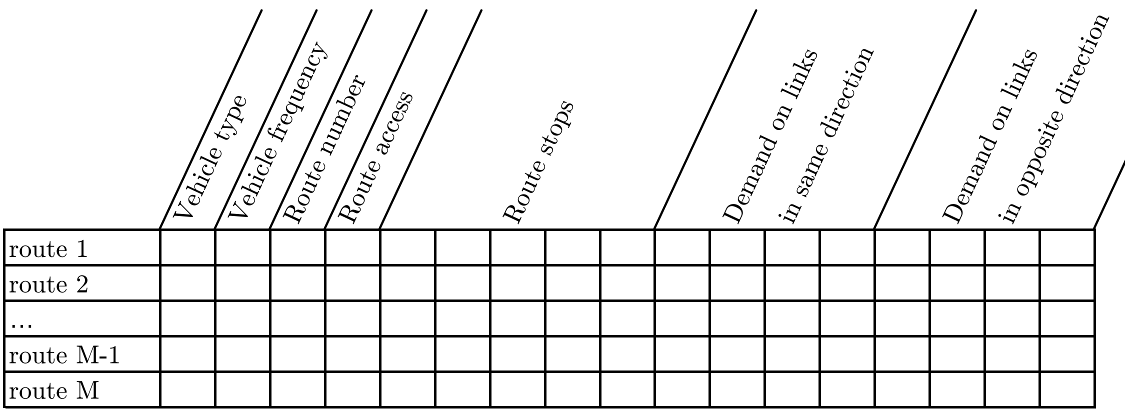


Figure 5.2: Representation of the decision matrix in *MATLAB*

Demand on links in opposite direction Indicates the amount of transit demand traversing the links on route m in the *opposite* direction to the sequence of the stops.

The route link assigned the highest transit demand is referred to as the *critical link*. The critical link is used to determine the frequency with which the vehicles will traverse route m. In the following sections the three procedures of the MNDA will be discussed in more detail.

5.2.1 The Initial Candidate Route Set Generation Procedure

The ICRSGP generates the solution space for the algorithm. The input to this procedure consists of: 1) the minimum and maximum route length, $L_{\min}$ and $L_{\max}$, respectively; 2) the distance matrix between the stops d_{ij}; and 3) either a matrix indicating which stops are directly connected c_{ij} or the maximum distance allowed between two connected stops $l_{\max}$. Figure 5.3 shows the outline of the ICRSGP.

The routes are generated using Dijkstras shortest path algorithm (Dijkstra, 1959) to determine the shortest route between all stops within the network. Once the shortest route has been identified, Yen's K-shortest path algorithm (Yen, 1971) is used to find alternative routes from the origin stop to the destination stop. This is done by sequentially removing each link in the shortest path from the solution space, forcing Dijkstra's shortest path algorithm to find another route. Should, for example, the shortest route between two stops consist of 12 links, then 12 alternative routes will be found. After all possible routes are generated, each route is subjected to the ROUTE LENGTH CONSTRAINT (5.9) to determine its validity. Additional constraints are that no two routes should overlap more than 70%, and the length of two routes between the same origin and destination stops should not differ by more than 150% (Chakroborty, 2003).

Note that the structure of the algorithm is responsible for creating a solution space in such a way that similar routes, with only a few links that differ, are stored next to one another. Once the solution space is generated a large set of possible routes exists for making up the network. A tool is needed to help us evaluate a subset of these routes which constitute a possible network. The NAP proposed in this dissertation is such a tool, as it assists us to evaluate the quality of a given solution.

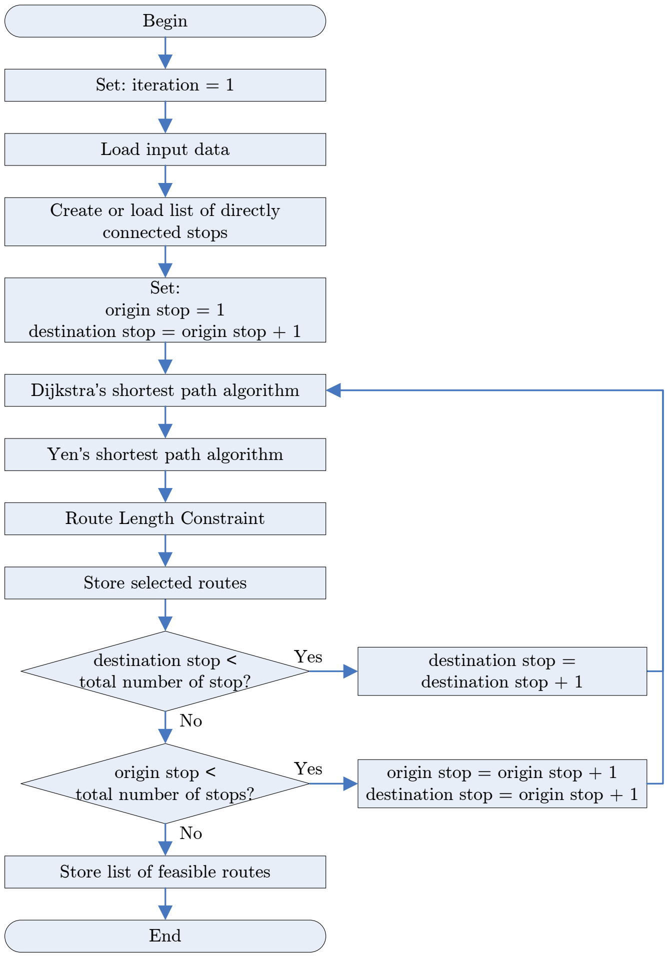


Figure 5.3: The Initial Candidate Route Set Generation Procedure

5.2.2 The Network Analysis Procedure

The main purpose of the NAP is to assign the transit demand to the appropriate links of the selected routes, and to set the vehicle frequencies required on each route in order to meet the demand effectively. Figure 5.4 shows the structure of the NAP. The input required for the NAP is listed below:

Distance Matix Matrix containing the distances between all the stops in the network. If two stops are not connected in the network, the value of infinity is used to indicate this. The distance matrix is developed in the ICRSGP.

Demand Matrix Matrix that contains the transit demand from every stop to every other stop in the network; provided by the user, or generated by combining the list linking each commuter point to a bus stop from the Bus Stop Placement Algorithm (BSPA), with the list of linked production and attraction points from the Population

Generation Algorithm (PGA).

Routes in the current network Provided by the SASP.

Vehicle type operating on each route Provided by the SASP.

Vehicle characteristics Seating capacity, load factor, and running costs; provided by the user.

Commuter related costs Average value of the commuter's time, penalty associated with route transfers, and unmet demand penalty; provided by the user.

For the purpose of describing the NAP, the structure of the algorithm is divided into five steps.

1) Initialization of the algorithm

After the input is loaded into the algorithm, the initial vehicle frequencies of each route are set to six vehicles per hour. This value will be updated once all the demand is allocated to the routes and the actual frequencies required to meet the demand have been determined in step 4. The algorithm will run through steps 2 and 3 for every OD pair if there is demand to be allocated.

2) Calculation of total travel time on all possible paths

The travel time is determined for every possible path between the origin and destination stop. These paths can contain zero, one, or two transfers between routes. The total travel time of a certain path is calculated by the sum of the waiting time at a stop, the in-vehicle travel time, and possibly the transfer penalty time. Should, for example, a path contain one transfer, then the sum is taken of the waiting time at origin stop, the in-vehicle travel time on origin route, the waiting time at the transfer stop, the in-vehicle travel time on destination route, and the transfer penalty time. The waiting time at a stop is based on the frequency with which the vehicles traverse that route. The in-vehicle travel time is based on the average speed of the vehicle operating on that route.

3) Route selection, demand assignment, and calculation of the user cost

The NAP checks if there are any viable paths available to which to assign the specific OD demand. If no path is available, the demand is assigned to the unsatisfied demand counter,

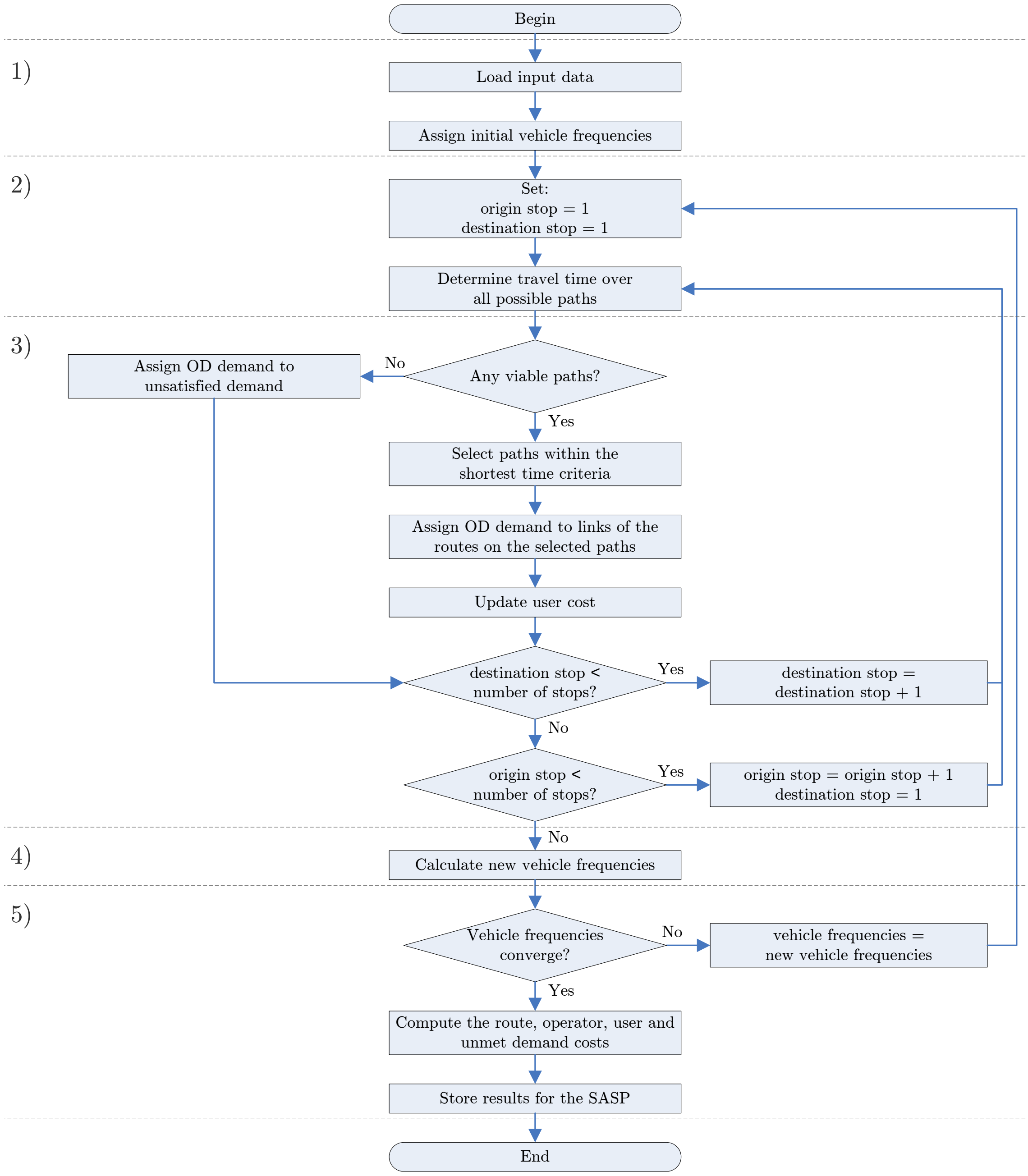


Figure 5.4: The Network Analysis Procedure

which means that the current network is not able to accommodate these commuters. This demand is penalized by a value provided by the user and is incorporated into the objective function. If paths are available to accommodate the demand, the fastest path will be selected. This path will function as a benchmark to determine which other paths will also be selected for demand allocation. The user can indicate the amount by which the travel time on the other paths can exceed the travel time on the fastest path, in order still to be selected for demand allocation. This can be, for example, a maximum of 30 percent longer travel time compared with the fastest path. The appropriate links of the routes on these selected paths will then be assigned the demand, in proportion to their travel time. Faster paths are assigned a larger proportion of the demand than slower paths. After the demand assignment the user cost is updated. The user cost is determined by multiplying the travel time on a certain path with the demand allocated to that path.

4) Calculation of vehicle frequencies

Once all the demand for every OD pair has been allocated, the vehicle frequency required to handle the demand on each route effectively is calculated. This is based on the demand assigned to the critical link of each route, which is the link with the highest amount of demand in one direction. The vehicle frequency is calculated by using (5.2) on page 49.

5) Checking for conversion of vehicle frequencies

Once the vehicle frequencies are calculated for each route, they are compared with the initial vehicle frequencies. If any of these values differ by more than 10 percent, the allocation of transit demand for every OD pair is repeated (steps 2 and 3) until there is internal consistency with the route frequencies. This is necessary because the amount of demand assigned to a route depends on the frequency of the vehicles operating on that route, since it influences the total travel time of the OD paths. Once there is internal consistency, the NAP is completed. The route, operator, user, and unsatisfied demand costs are computed and sent to the SASP, together with the route utilization values.

5.2.3 The Simulated Annealing Search Procedure

The SASP interfaces with the solution space generated by the ICRSGP in order to propose a change to the network, and then uses the NAP to evaluate the proposed change before deciding either to accept or to reject the change. This process is repeated for a given number of iterations in search of better solutions while avoiding local optima. The following input is required by the SASP:

Routes and route distances Represents the solution space. This matrix is developed by the ICRSGP.

Simulated Annealing (SA) cooling schedule parameters The parameters that guide the temperature of the SASP. These are the temperature objective factor, the temperature reduction factor, the number of iterations performed at each temperature, and the minimum temperature needed to be reached until the algorithm stops. Parameter values are provided by the user.

Change parameters The parameters that guide the change component of the algorithm. These are the route change probability (indicating the probability of initiating a route change over a vehicle change), and the local move probability (indicating the probability of initiating a local move over a global move). Value for these parameters are provided by the user.

User input Other input provided by the user, such as the number of routes in the network M and the relative weights of the costs W_1, W_2 and W_3.

The structure of the SASP is presented in Figure 5.5. For the purpose of describing the algorithm, the structure is divided into four steps.

1) Initialization of the algorithm

After the input is loaded, the algorithm constructs an initial solution consisting of routes selected from the solution space and arbitrary vehicle types on each route. The NAP is then called to evaluate the initial solution and calculate the objective function value. On this initial objective function value, the initial temperature of the SASP is set, by multiplying the objective function value by a predetermined value.

2) Selecting the type of change

The SASP tries to improve its current solution by making an intelligent change and then evaluating the effect of that change on the network. This change can be a change either to the routes of the current solution or to the vehicle type operating on the routes. The probability of choosing a certain type of change is based on a value provided by the user.

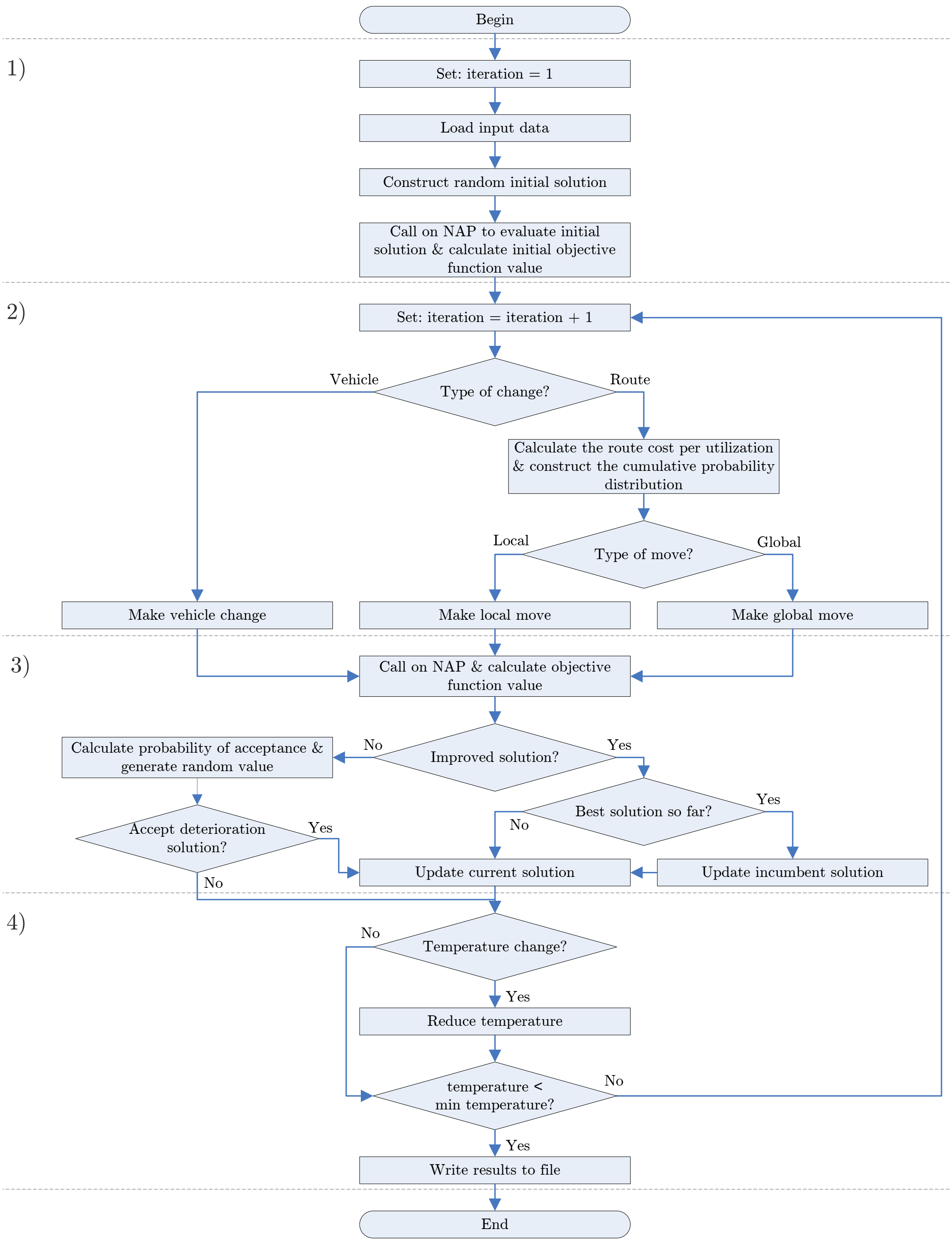


Figure 5.5: The Simulated Annealing Search Procedure

If a route change is initiated, a route is picked from the current solution. The initial random route change procedure resulted in slow convergence rates and low repeatability. Therefore, an alternative procedure was developed that increases the intelligence of the search algorithm. The NAP assigns demand based on travel time, which means that a route with high utilization is beneficial for the network. The total cost of all the routes is represented in the objective function value, which means that a route associated with a low cost is also beneficial to the network. However, high utilization and low cost are usually two forces working in opposite directions, since every commuter utilizing a certain route increases the cost associated with that route. By dividing the cost of a route by its utilization, a measurement value is created that indicates the cost of transporting one commuter on that route. The lower this value relative to the other routes, the better the relative rating of the route. So the SASP uses this value to develop a cumulative probability distribution in such a way that routes with a higher cost per utilization have a higher chance of being selected for a route change. Once a certain route is selected for a route change, either a local or a global change is initiated. The probability of choosing either of the changes depends on a value provided by the user. Since the solution space is ordered in such a way that similar routes are stored next to one another, the possibility is created to choose between a subtle change or a more drastic one. A local change causes a subtle change to the network, by picking a route moving at most 5 positions up or down from the current route's position in the solution space. A global change causes more radical changes to the network, by moving through between $[-0.5; 0.5]$ of the solution space, depending on a random number.

If a vehicle change is initiated, a certain route will be selected on which the change will occur. Initially the vehicle was randomly changed on a randomly chosen route. However this often did more damage than good. Routes with either the highest or the lowest vehicle frequencies benefit the most from vehicle changes. Routes with a high vehicle frequency benefit mostly from increasing the capacity of the vehicles, as this reduces the operator cost while barely affecting the user cost. Routes with low vehicle frequencies benefit mostly from changing to smaller vehicles, as this increases the frequency thus ensuring faster travel time for its commuters. The vehicle change procedure was therefore changed to one where the route with either the highest or the lowest vehicle frequency is chosen. If the route with the highest vehicle frequency is chosen, the current vehicle type is replaced by a slightly bigger vehicle type, if possible. Similarly, if the route with the lowest frequency is chosen, the current vehicle type is replaced by a slightly smaller vehicle type. Although routes are no longer randomly selected, sufficient diversification is still ensured since routes of the current solution are constantly changed, resulting in constantly changing vehicle types and frequencies.

3) Evaluation of the proposed change

Once a change has been made, the NAP is initiated to evaluate the effect of the change. From the results of the NAP, the objective function value is calculated. If the solution is an improvement, the change is accepted; and if the solution is also the best solution so far, the incumbent solution is updated. If the solution was not an improvement, the change will be accepted with a certain probability depending on the amount of deterioration and the temperature of the SA cooling schedule.

4) Reducing the temperature

The temperature of the SA cooling schedule is reduced after a predetermined number of iterations, making the probability of accepting deteriorating moves smaller and smaller. Steps 3 and 4 are repeated until the temperature reaches a predetermined minimum level. The routes, route frequencies, vehicle types, and objective function values of the incumbent solution network are then written to file.

5.3 Algorithm performance

Before the algorithm is used to design a new network for the whole City of Tshwane Metropolitan Municipality (CTMM), it is tested on a small test network. In this section the outline of the test network is presented and the input of the algorithm discussed. For the input parameters related to the SA cooling schedule and the change component, it is unclear what the appropriate value of the parameter should be to produce the best possible solution network. Therefore, a sensitivity analysis is done on a test network. Using the appropriate values found through the sensitivity analysis, the results of the algorithm are presented.

5.3.1 Outline of test network

The MNDA is tested on the network shown in Figure 5.6 used by Pattnaik et al. (1998) and Ngamchai and Lovell (2003).

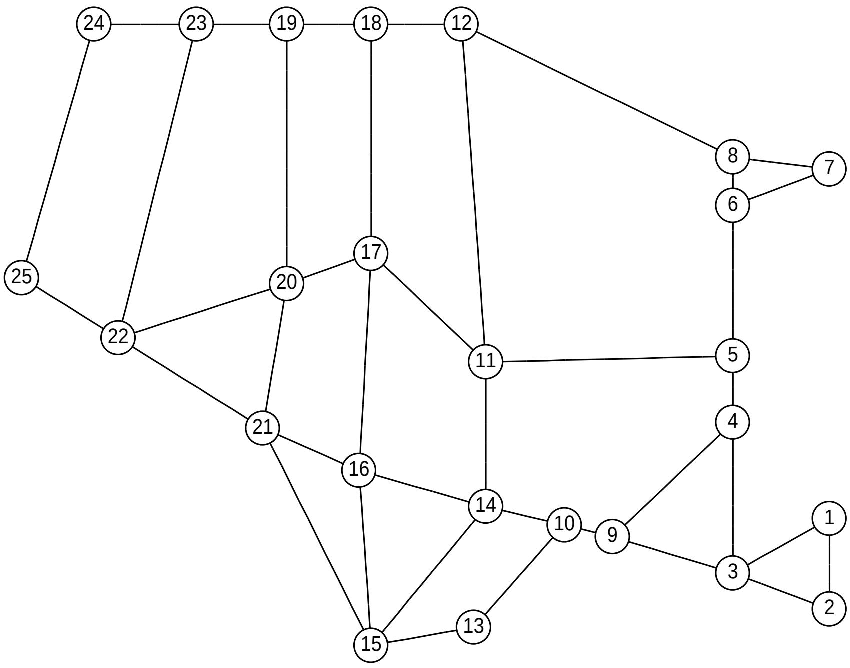


Figure 5.6: Configuration of network used to test the MNDA. Adopted from Ngamchai and Lovell (2003).

The same OD demand data is used as in the study of Ngamchai and Lovell (2003) and is presented in Appendix D, together with the coordinates of the stops.

For the purpose of testing the algorithm, three bus types are used: the minibus-taxis, and single- and double-decker buses. The characteristics of these buses are listed in Table 5.3.

Table 5.3: Characteristics of the vehicle types used to test the algorithm

Vehicle type	**Seats**	**Max load factor**	**Rand/km**	**Driver cost**	**Speed**
1	101	1.0	R 9.00	R 58.00/h	45 km/h
2	54	1.3	R 8.15	R 58.00/h	45 km/h
3	22	1.0	R 4.30	R 58.00/h	50 km/h

The maximum and minimum length allowed for any of the routes, L_{max} and L_{min}, is set at 10 km and 20 km respectively. The maximum distance allowed between two directly connected stops, l_{max}, is not specified since the direct connection between the stops is entered manually, according to the links shown in Figure 5.6.

The per-hour value of the commuter's time spent travelling, C, is set at ZAR 37.42. This value is based on the average hourly income of employed persons in Tshwane in Rands (Joburg, 2007). The value for the cost for unsatisfied demand, U, is arbitrarily set at ZAR 1000. The amount chosen tends to move towards networks where all the demand is met. Since the purpose of this project is to promote public transport, this high value is justified.

For the transfer penalty T a value of 5 minutes is chosen, similar to Fan and Machemehl (2004). The weights reflecting the relative importance of the three components of the objective function W_1, W_2 and W_3 are also based on the work of Fan and Machemehl (2004), and set to 1, 1, and 0.5 respectively.

The rest of the input to the algorithm is the parameters that guide the metaheuristic. These parameters influence the search behaviour of the algorithm and need to be fine-tuned in order to optimize the search. Therefore a sensitivity analysis is done to find the most appropriate value for these parameters. The parameters are presented in the next section.

5.3.2 Sensitivity analysis on test network

For the sensitivity analysis seven parameters are chosen, of which the first four are related to the SA cooling schedule. Each parameter is briefly discussed below.

tempReductionFreq Determines after how many iterations the temperature is reduced.

tempReductionFactor Determines by how much the value of the temperature is reduced.

tempObjectiveFactor Used to calculate the initial temperature, by multiplying this parameter by the initial objective function value.

minTemp Used as the stopping criterion of the algorithm. If the temperature reaches the value of *minTemp* the search for a better solution is complete.

routeChangeProbability Determines the probability that the algorithm will choose a route change over a vehicle change.

localMoveProbability Determines the probability that a local move is chosen over a global move.

numberOfRoutes Determines the number of routes used in the search.

For each parameter five values are chosen and tested during the sensitivity analysis (see Table 5.4). The number of possible parameter combinations, 7^5, is too high for them all to be tested, so a baseline is sought of a reasonably good solution from which the effect of the parameter values on the objective function value is determined.

Table 5.4: Parameter values chosen for the sensitivity analysis of the MNDA

Parameter	**Values**	**Parameter**	**Values**
tempReductionFreq	5, 15, 25, 35, 45	routeChangeProb	0.1, 0.3, 0.5, 0.7, 0.9
tempReductionFactor	0.75, 0.8, 0.85, 0.9, 0.95	localMoveProb	0.1, 0.3, 0.5, 0.7, 0.9
tempObjectiveFactor	0.05, 0.1, 0.5, 1, 5	numberOfRoutes	5, 7, 9, 11, 13
minTemp	5, 10, 50, 100, 1000		

To determine the baseline, 50 parameter settings chosen randomly are compared by their average objective function value and their Mean Absolute Deviation (MAD) %. The values used for each parameter setting are provided in Appendix E. The runs are limited to a time period of 48 hours, since a parameter combination is sought that produces good results within a reasonable time period. To create a better overview, the results of only 45 of these combinations are shown in Figure 5.7, since 5 results were largely worse than the 45 shown. The best objective function value is produced by using combination 7, and the best MAD % by combination 39. The most important objective is to find a combination that produces good results, therefore combination 7 is chosen as the baseline for the next step of the sensitivity analysis. The associated parameter values with this combination are shown in Table 5.5.

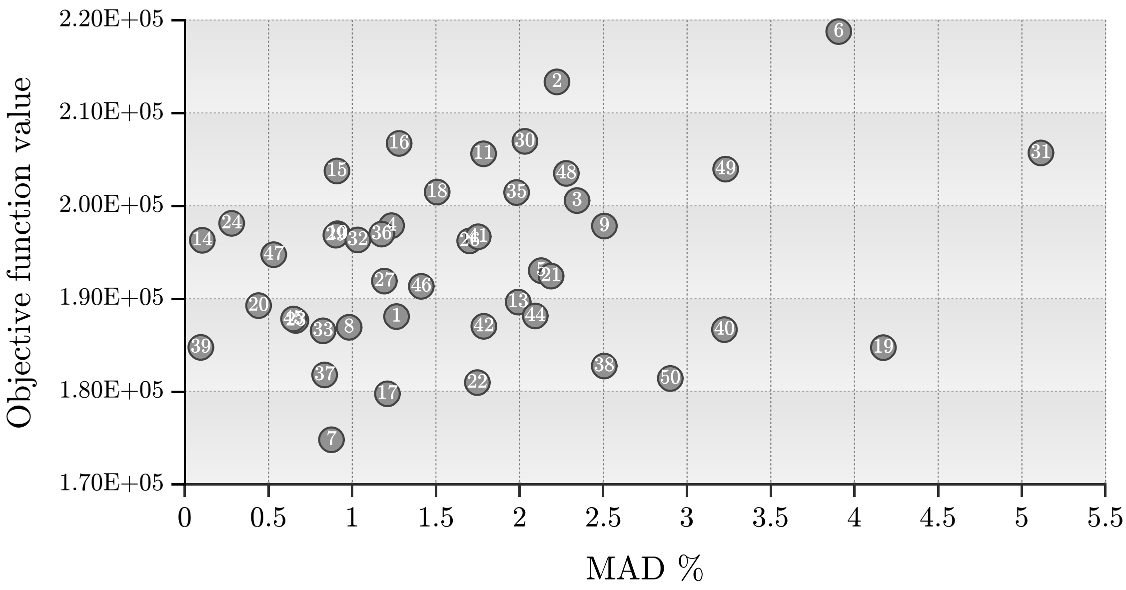


Figure 5.7: Results of 50 different parameter settings; results of only 45 fall within the figure's extent. Each point on the graph represents the average objective function value and MAD % retrieved after three runs of one parameter setting. The number on each point in the graph refers to the ID-number given to each parameter setting, and can be used to find the corresponding parameter values of each point through Appendix E.

Table 5.5: Values used as the baseline parameter setting for the sensitivity analysis

Parameter	**Value**	**Parameter**	**Value**
tempReductionFreq	25	routeChangeProb	0.7
tempReductionFactor	0.95	localMoveProb	0.3
tempObjectiveFactor	0.5	numberOfRoutes	9
minTemp	5		

The results of the sensitivity analysis on the seven parameters are shown in Figure 5.8, where the average objective function value and the MAD % are compared after each combination is run five times. The effect of the values of one parameter on the objective function value are evaluated while the values of all the other parameters are kept constant by assigning them the baseline values as presented in Table 5.5.

The *tempReductionFreq* parameter shown in Figure 5.8(a) produces the best results with its value set at 25. The reason higher values don't produce better results—as was

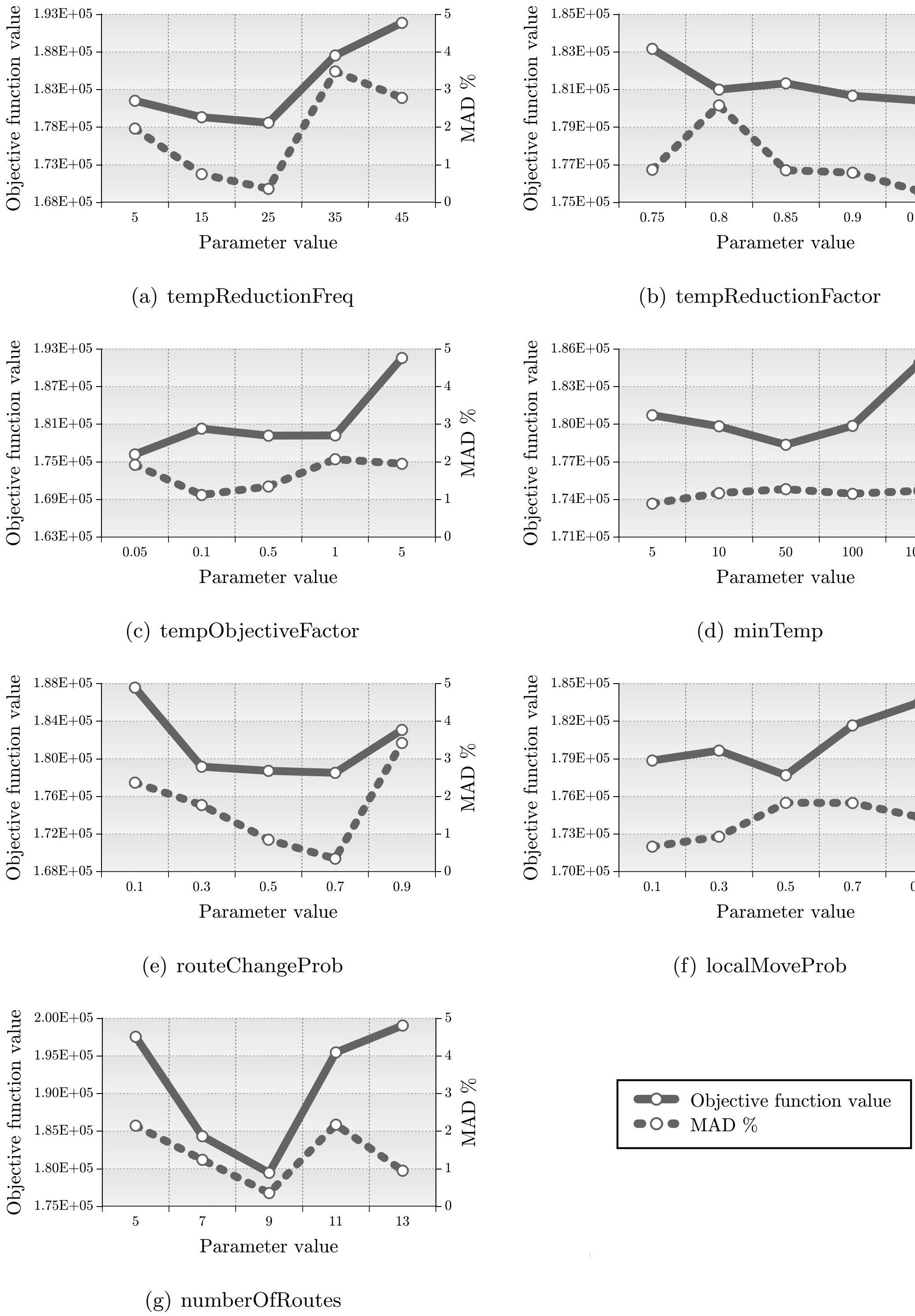


Figure 5.8: Sensitivity analysis on seven parameters of the MNDA

initially expected—could be caused by the maximum run time constraint of 48 hours. Higher values increase the run time of the algorithm, since the temperature is reduced less frequently. For the *tempReductionFactor* parameter the best results were produced with a value of 0.95, indicating a slow decrease of the temperature is beneficial for the algorithm (Figure 5.8(b)). The *tempObjectiveFactor* parameter in Figure 5.8(c) produces the best objective function values with its value set at 0.05, and the best MAD % with a value of 0.1. The preference for low values indicates that to produce good results the initial temperature can be quite low—something that was initially not anticipated. This could however also be caused by the maximum run time constraint. The value 0.05 is chosen for this parameter. In Figure 5.8(d) the *minTemp* parameter shows the best results with a value of 50, which again could be explained by the maximum run time constraint. The probability of choosing a route-change over a vehicle-change produces similar results between 30% and 70% in Figure 5.8(e), which indicates that good results are produced as long as one type of change is not predominally preferred. The MAD % however shows the best results with a parameter value of 0.7, and therefore this value is chosen for the *routeChangeProbability* parameter. For the *localMoveProbability* parameter in Figure 5.8(f) the best objective function values are retrieved with a value of 0.5, providing equal chance of choosing a local-move or global-move. The results of the *numberOfRoutes* parameter in Figure 5.8(g) shows that the best solutions are found with networks containing 9 routes.

5.3.3 Proposed network

The proposed network is generated by using the parameter values retrieved through the sensitivity analysis, as presented in Table 5.6. Ten runs are performed using this parameter combination, without the maximum run time constraint of 48 hours. The best solution is presented in Figure 5.9.

Table 5.6: Parameter values retrieved through the sensitivity analysis

Parameter	**Value**	**Parameter**	**Value**
tempReductionFreq	25	routeChangeProb	0.7
tempReductionFactor	0.95	localMoveProb	0.5
tempObjectiveFactor	0.05	numberOfRoutes	9
minTemp	50		

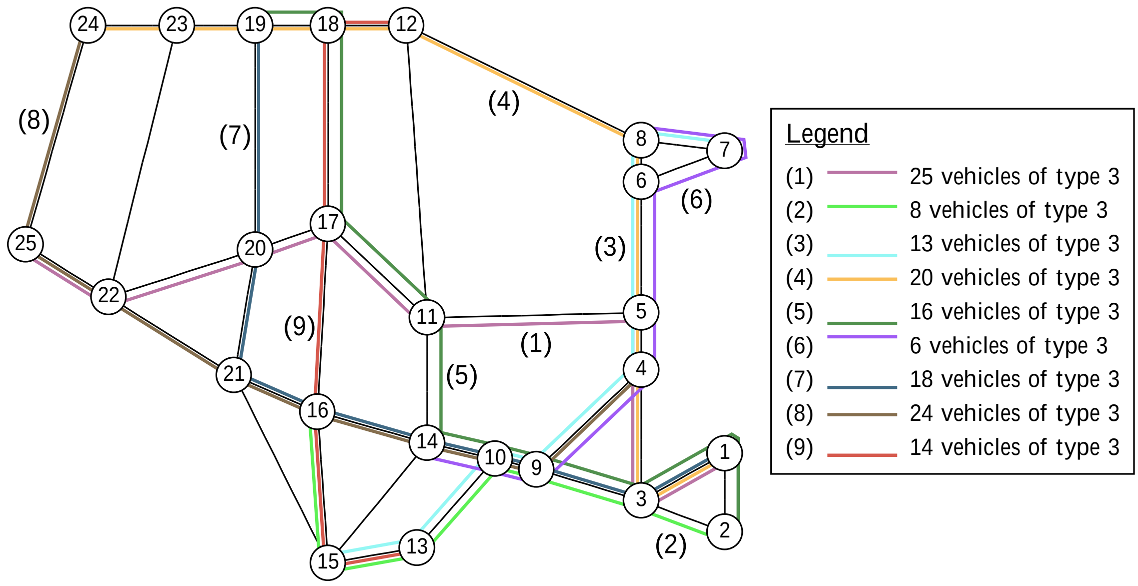


Figure 5.9: Proposed network by the MNDA

Interesting about the result is that all the routes in the network are assigned vehicles of type 3: the minibus-taxi. Even for routes assigned up to 25 vehicles, the model still prefers minibus-taxis over larger buses—something that was not initially anticipated. It is assumed that this is caused by the superior travel speed and increased vehicle frequencies on routes that are assigned minibus-taxis. Although the operating cost per-commuter of minibus-taxis is higher than the operating cost of larger buses, the total travel time is reduced on routes with minibus-taxis due to faster travel speed and higher frequencies (which decrease the waiting time for commuters). The reduced total travel time has apparently a larger impact on the objective function than the increased operator cost. Thus it seems that from a total cost perceptive a network containing only minibus-taxis in beneficial for both the operator and commuter.

Although the results strictly only apply to the test network, a refection of the results on the current network of the CTMM is beneficial. You may wonder, for example, why currently larger buses are still operating in the CTMM, assuming that the model is correct. To answer this question two potential causes are identified and discussed below.

The larger bus operators are currently subsidized by the national and local authorities. For some operators these subsidies amount up to 65% of their annual turnover. To investigate the effect of these subsidies on the model, several runs are performed using different subsidy values for the larger buses, ranging from 10% to 65%. The results show that with a subsidy value of 50% the vehicle types 2 and 3 are divided amongst the routes by a ratio of 2/3 to 1/3, respectively. With a subsidy value of 65% all the routes are assigned a vehicle of type 2: the bus with 54 seats. This shows that the model is also capable of producing networks similar to the current network of the CTMM. This however also raises the question that the assignment of the current subsidies should be re-evaluated.

The second identified cause is related to the size of the minibus-taxi. The model uses the newest type of minibus-taxi, capable of transporting 22 commuters. Currently however most minibus-taxis used in the CTMM are still of the older type, capable of transporting only 15 commuters. It is assumed that changing the seating capacity of the minibus-taxis in the model would make this type less favourable, since this will increase the per-commuter operating cost of the minibus-taxi.

The best solution, with an objective function value of 163 081, was found during a run of 3651 iterations that took 71 hours to complete on one node of the *Velocity* cluster (3 GHz processor). The solutions during the initial iterations have high objective function values—and are therefore not shown in Figure 5.10—but better solutions with objective function values around 200 000 are quickly found. The graph shows the typical characteristics of an SA metaheuristic: initially a broad search through the solution space to ensure divergence, and as the temperature decreases the solution slowly converges to the best local optimum found.

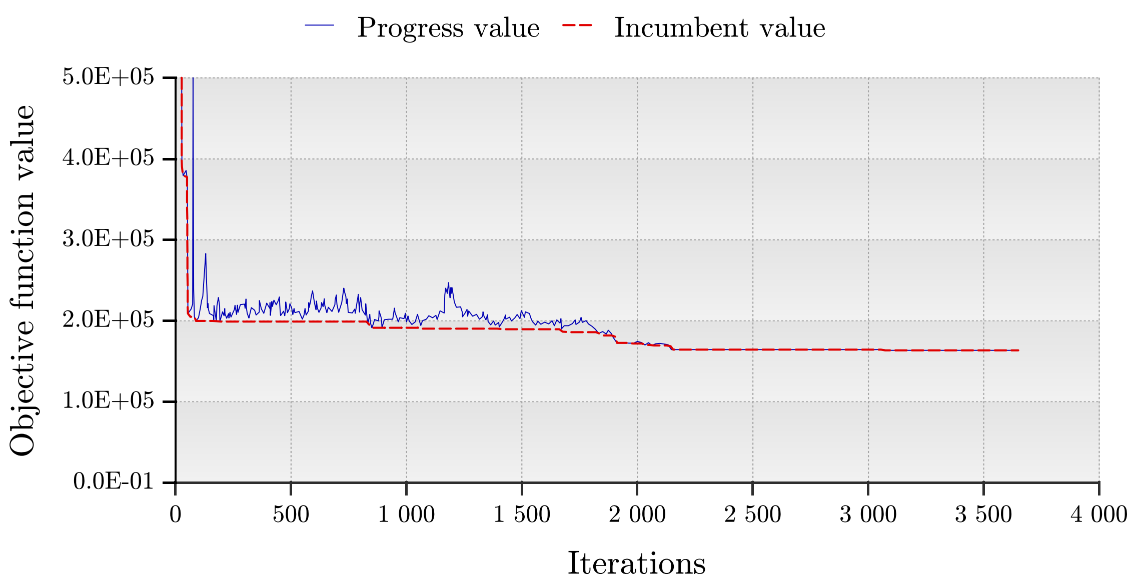


Figure 5.10: Progress of the algorithm during one run

5.4 Conclusion

In this chapter a new multimodal network design model is proposed based on the work of Fan and Machemehl (2006b). The proposed model consists of three procedure: the ICRSGP, the NAP, and the SASP. The ICRSGP creates the solution space according to the user defined feasible constraints. The NAP evaluates the current solution proposed by the SASP, by assigning the demand to the routes, determining the vehicle frequencies, and computing the performance measurements. The SASP is an SA based metaheuristic that applies changes to the current solution during each iteration, and then decides whether to accept or reject the proposed change after the evaluation of the solution by the NAP.

The MNDA is tested on a test network, for which the most appropriate values of seven parameters were determined by a sensitivity analysis. In order to finish the complete sensitivity analysis in the available amount of time, the run time was limited to 48 hours. The values retrieved from the sensitivity analysis are therefore not necessarily the best parameters to find a good solution, but the best parameters to find a good solution in an acceptable time window. The proposed solution for the test network shows that the MNDA is capable of designing good networks with multiple vehicle types.

Chapter 6

Model applied to the CTMM

To test the ability of the model to redesign a large-scale public transport network, the model is applied to the City of Tshwane Metropolitan Municipality (CTMM). The process followed by the model to transform the initial input data into an optimized network is divided into five steps, each discussed chronologically in the following sections.

6.1 Data into GIS

The appropriate geographic input data is identified, loaded into the Geographic Information System (GIS), manipulated where necessary, and stored in the geodatabase. The output of this step is a geodatabase similar to the design proposed in Section 3.1.2, consisting of all the data of the CTMM needed by the three main algorithms. The elements of the geodatabase are shown in Figure 6.1.

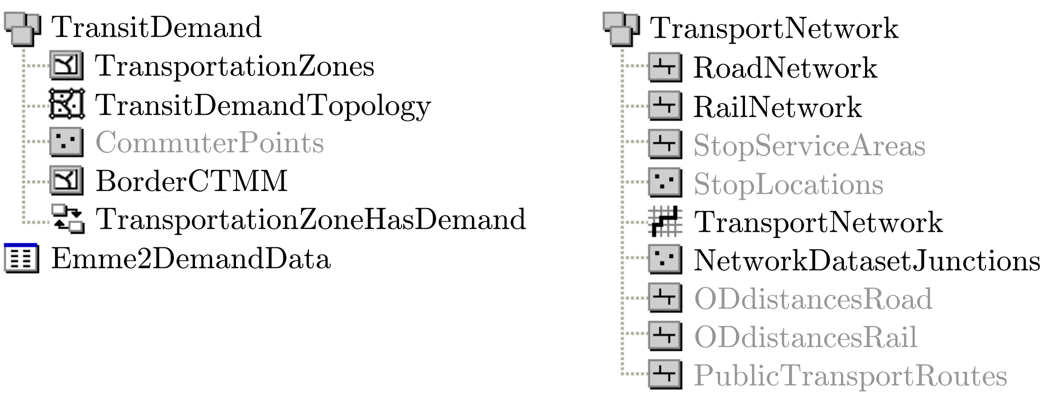


Figure 6.1: Elements of the geodatabase representing the geographic data of the CTMM. The elements displayed in black are the initial input data, and the elements displayed in grey are the elements produced in the next steps of the model.

To provide an overview of the input data of the CTMM, the *RoadNetwork*, *RailNetwork*, and *TransportationZones* feature classes are shown in Figure 6.2.

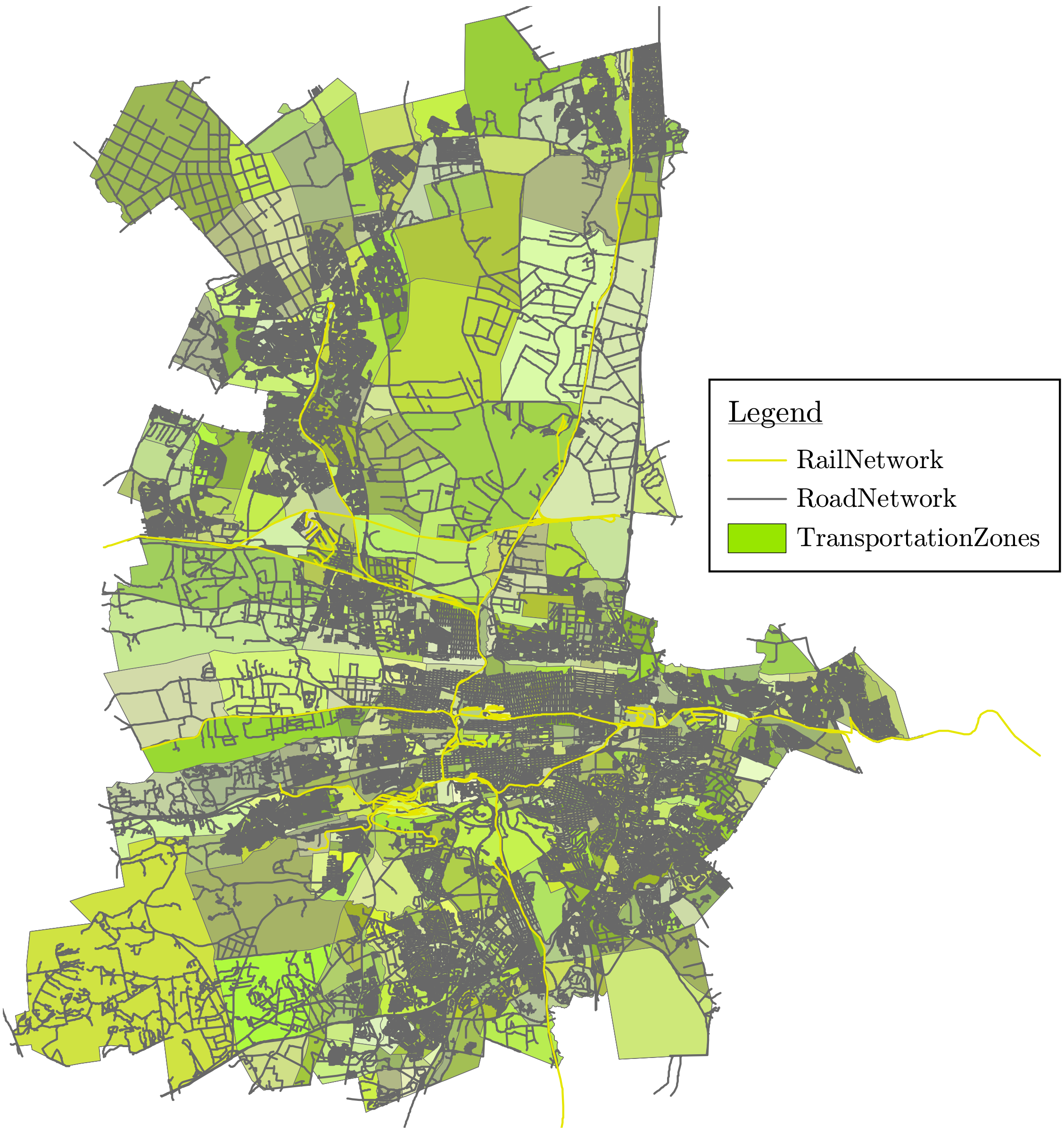


Figure 6.2: Overview of the geographic input data of the model for the CTMM

6.2 Population generation

The population generation is done by the Population Generation Algorithm (PGA). The input data for the PGA consists of the *Emme2DemandData* table and the *TransportationZones* feature class. For the purpose of this project, each element of the Origin-Destination (OD) demand matrix is divided by 20, to create a matrix representing 5% of the total demand in the CTMM. The main reason for considering only 5% of the total demand is to reduce the computational resources needed to run the Bus Stop Placement Algorithm (BSPA) and Multimodal Network Design Algorithm (MNDA). The output of the PGA, stored in the geodatabase as the *CommuterPoints* feature class, is shown in Figure 6.3 together with the *BorderCTMM* feature class.

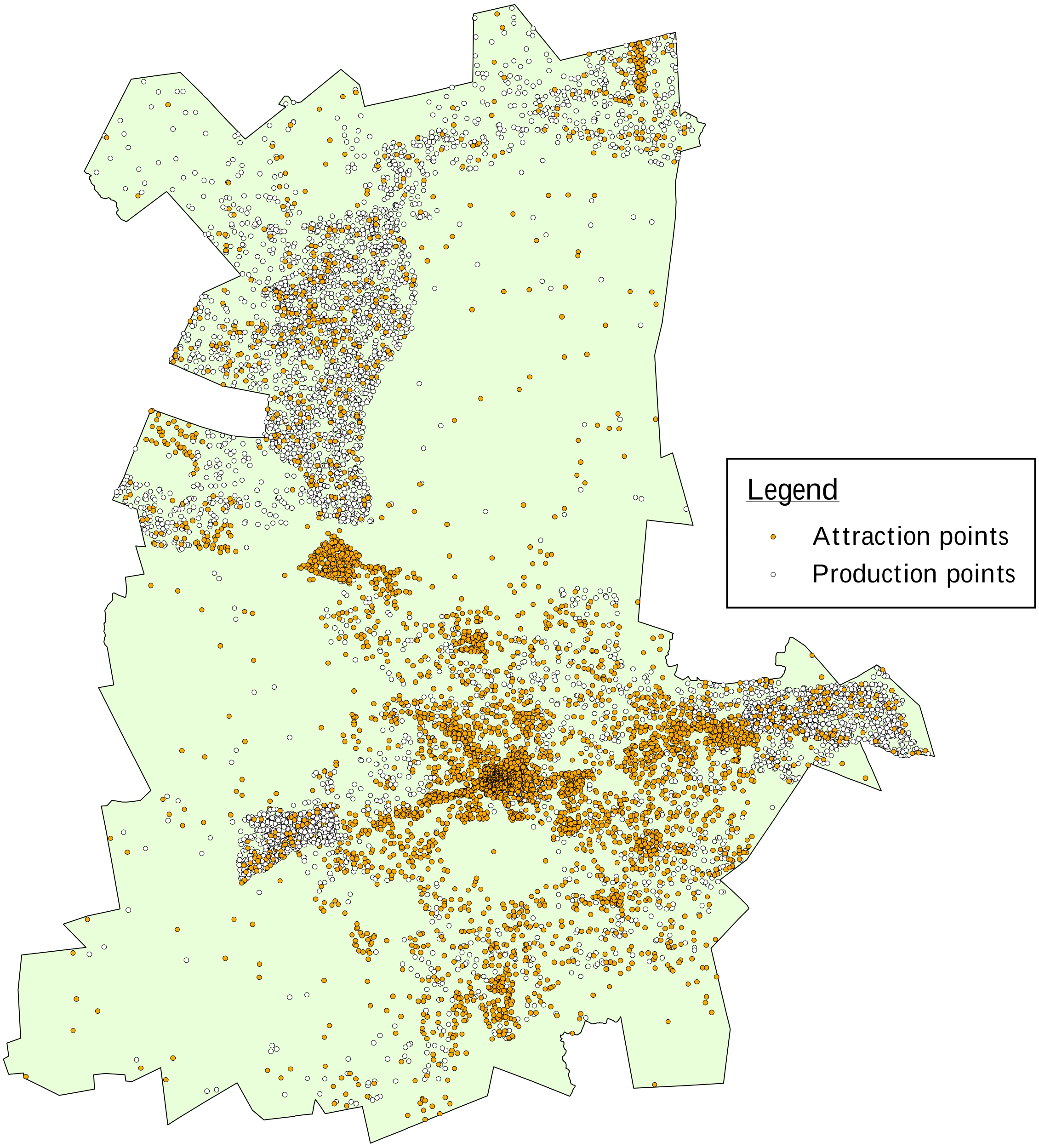


Figure 6.3: Placement of commuter points in the CTMM

6.3 Bus stop placement

The input data for the BSPA consists of the *CommuterPoints*, the *BorderCTMM*, the *RoadNetwork*, and the *RailNetwork* feature classes retrieved from the geodatabase. The *RailNetwork* feature class provides the algorithm with 77 fixed bus stops positions, which represent the bus stops at the train stations of the CTMM.

The weights of the objective function, W_1 and W_2, are set to 1 and 3.5 million respectively. The same values can be used as retrieved from the efficient frontier evaluation in Chapter 4, since the demand still represents 5% of the total demand.

Although values retrieved through a sensitivity analysis are known to be problem specific, as indicated by Fan and Machemehl (2006b), the same parameter values proposed in Chapter 4 are used for the seven parameters that guide the search of the BSPA. The parameter combination used in this step is shown in Table 6.1.

Table 6.1: Values selected for the seven parameters guiding the BSPA

Parameter	**Value**	**Parameter**	**Value**
tempReductionFreq	45	initialNrStops	40
tempReductionFactor	0.95	maxPossibleMoves	100
tempObjectiveFactor	5	perburbationFreq	50
minTemp	5		

The BSPA is run 10 times on the *Velocity* cluster, and the best result is selected and stored in the geodatabase. The positions of the bus stops, after being aligned with the road network, are shown in Figure 6.4. These positions were retrieved during a run of 144 hours.

6.4 Network design

Before the MNDA is able to run, the OD distances are generated by the GIS. Since the train is included as a mode of transport, the OD distances are generated by using both the road and the rail network. The 77 fixed stops will provide commuters with the ability to transfer between the road and the rail network—with each fixed stop representing a bus stop and a train station. The OD distances using only the road network are generated for all the stops, and the OD distances using only the rail network are generated for only

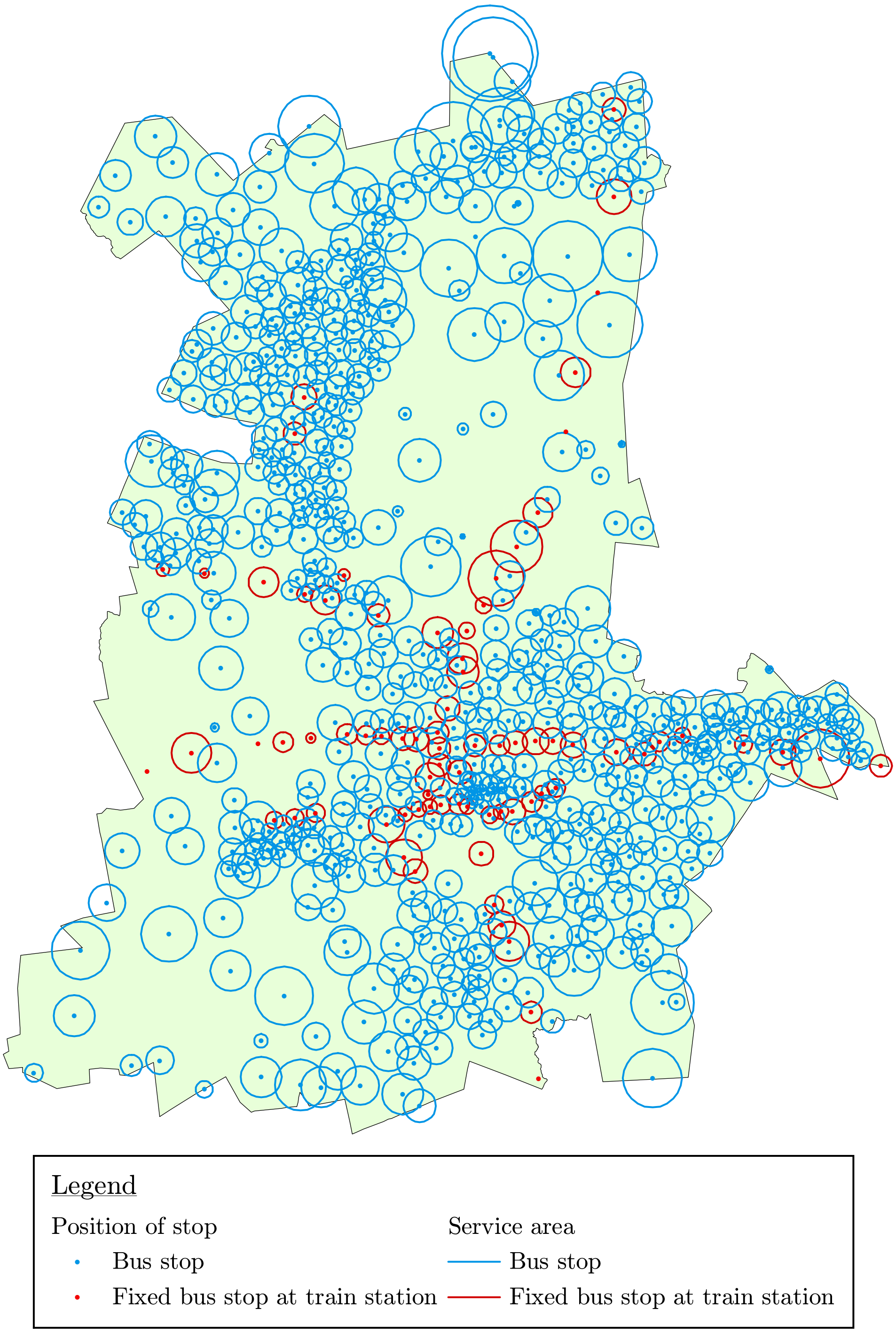


Figure 6.4: Placement of bus stops in CTMM

the fixed stops. The OD distances are stored in the geodatabase as the *ODdistancesRoad* and *ODdistancesRail* feature classes.

The demand data is generated by combining the list linking each commuter point to a bus stop from the BSPA, with the list of linked production and attraction points from the PGA. The resulting values are multiplied by 20 to convert the 5% sample to demand data representing 100% of the total demand. The resulting OD demand matrix is stored for the MNDA.

Since the train is added as a mode of transport for the network design of the CTMM—compared to the test network used in Chapter 5—a fourth vehicle type is introduced. The characteristics of all the four vehicle types are listed in Table 6.2.

Table 6.2: Characteristics of the vehicle types used to test the algorithm

Vehicle type	**Seats**	**Max load factor**	**Rand/km**	**Driver Cost**	**Speed**
1	101	1.0	R 9.00	R 58.00/h	45 km/h
2	54	1.3	R 8.15	R 58.00/h	45 km/h
3	22	1.0	R 4.30	R 58.00/h	50 km/h
4	1730	1.3	R 105	R 58.00/h	25 km/h

The minimum and maximum length allowed for any route in the network, $L_{\min}$ and $L_{\max}$, is set at 10 km and 30 km respectively. The maximum distance allowed between two directly connected stops, $l_{\max}$, is initially set at 5 km. Only for the stops that do not connected to any other stop after the first try with $l_{\max}$ set at 5 km, the value for $l_{\max}$ is iteratively increased by 5 km until all the stops are connected to at least one other stop. The value for the cost for unsatisfied demand, U, is increased—compared to the test network—to an arbitrarily value of 1 million ZAR, to compensate for the increase of the average travel time of commuters, forcing the algorithm to produce a network with no unmet demand. The values for the per-hour commuter's time spent travelling, C, the transfer penalty, T, and the objective function weights, W_1, W_2, and W_3, are all adopted from the runs on the test network and set at 37.42 ZAR, 5 minutes, 1, 1, and 0.5 respectively.

Similar to the bus stop placement parameters, the same values are used as retrieved from the sensitivity analysis on the test network in Chapter 5, except for the *numberOfRoutes* parameter. The number of routes in the network should obviously be higher, due to the increase of the size of the network. Several test runs were initiated, and showed that the most appropriate number of routes should be around 250. The parameter combination used to run the algorithm on the CTMM is presented in Table 6.3.

The best result of the MNDA is stored in the geodatabase as the *TransportRoutes*

Table 6.3: Values selected for the seven parameters guiding the MNDA

Parameter	**Value**	**Parameter**	**Value**
tempReductionFreq	25	routeChangeProbability	0.7
tempReductionFactor	0.95	localMoveProbability	0.5
tempObjectiveFactor	0.05	numberOfRoutes	250
minTemp	50		

feature class. Due to time limitations the runs were limited to 1224 hours, which proved insufficient time for the algorithm to converge to a good local optimum solution. This is illustrated in Figure 6.5, where the objective function value of the current solution and the incumbent solution during each iteration are shown.

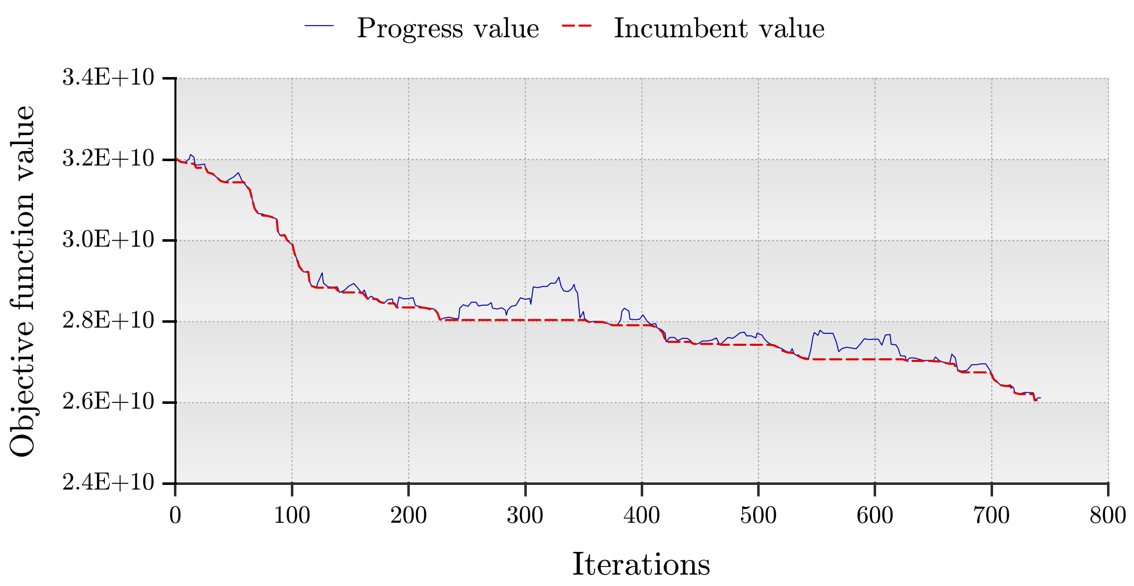


Figure 6.5: Progress of the algorithm during one run

The final solution, retrieved after 742 iterations, results in an objective function value of 26.05 billion. The solution presented in Figure 6.6 shows that the algorithm is capable of positioning the routes in the areas where the demand is the highest, and of creating a network that provides public transport for a large portion of the commuters. The algorithm however, needs substantial computational resources in order to find a near-optimal solution. Even in the broad time frame given, the algorithm is not capable of converging to a solution with an unmet demand of zero. The efficiency of the MNDA should be improved or the run time should be extended, in order for the algorithm to find better solutions.

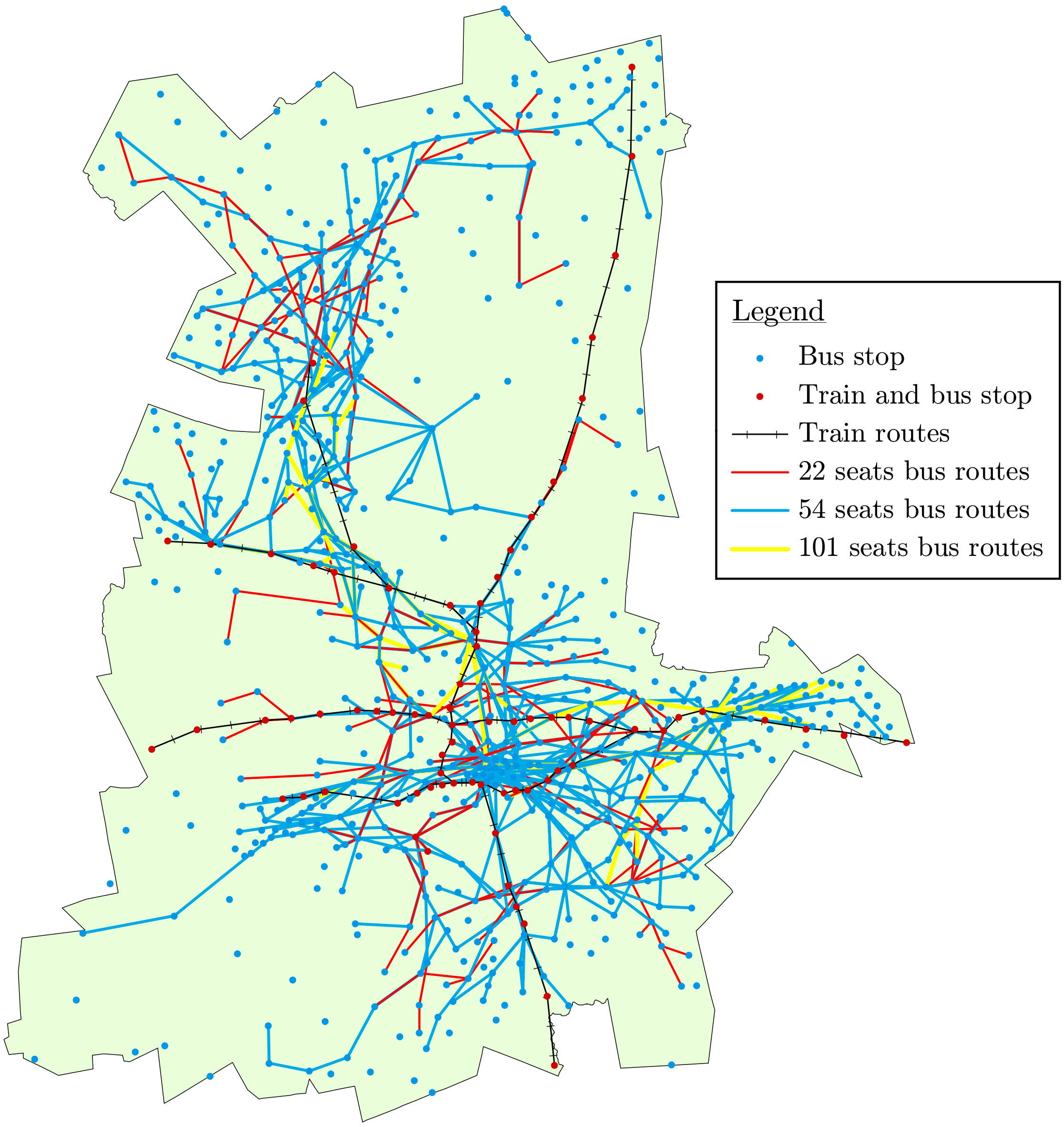


Figure 6.6: Network design for the CTMM

6.5 Proposed network

The eventual result of the model applied to the CTMM is presented in Figure 6.7 by combining the main output of the three algorithms into one figure. This consists of the *CommuterPoints*, *StopLocations*, and *TransportRoutes* feature classes produced by the

PGA, BSPA, and MNDA respectively. The results show that the model is capable of designing a new network for a large metropolitan area such as the CTMM. Unfortunately, due to time limitations, the potential of the MNDA component has not been fully utilized.

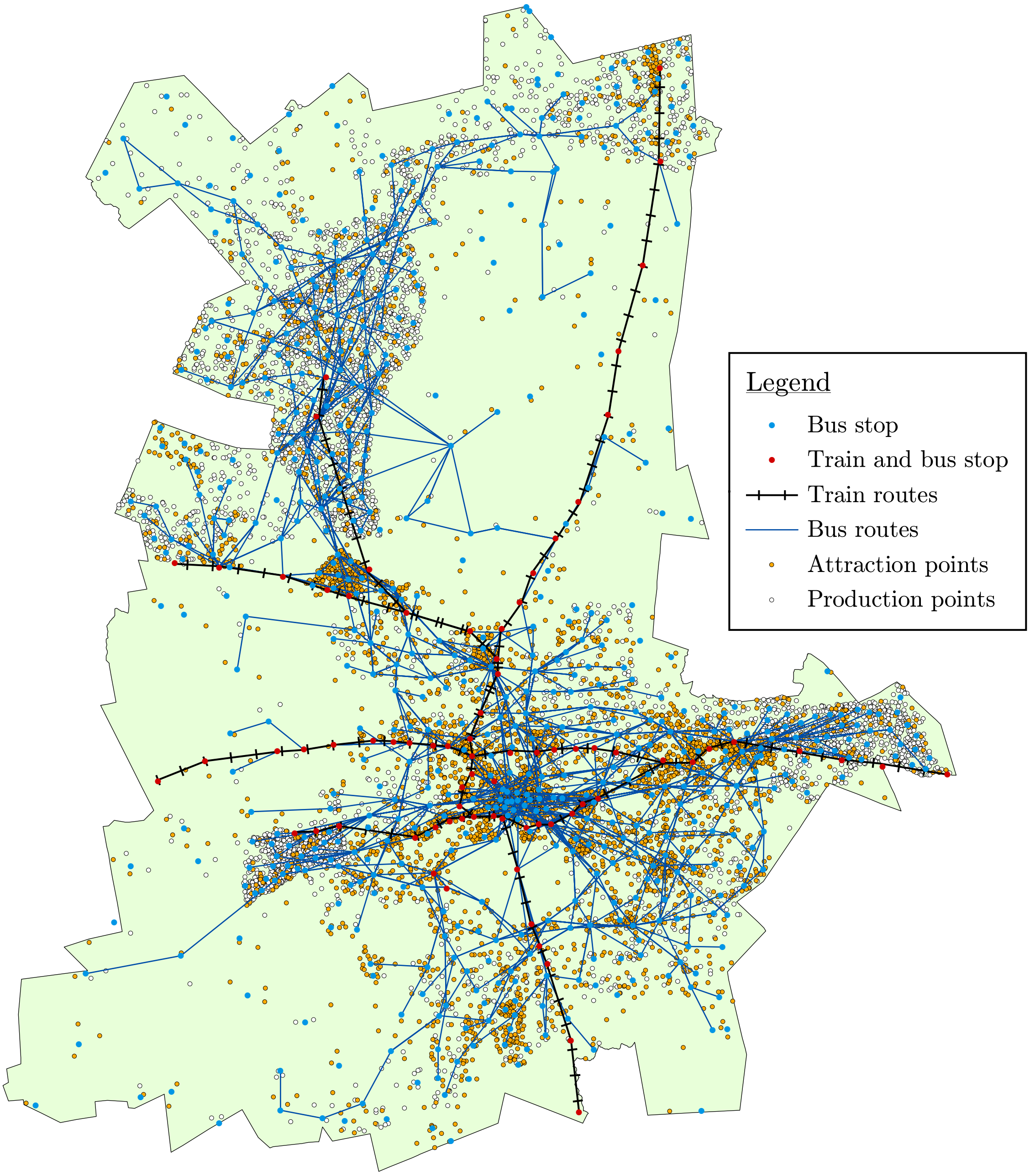


Figure 6.7: Proposed solution for the CTMM

Chapter 7

Conclusions and recommendations

7.1 Conclusions

The answer to the project's research question, "How can a fully-operational network design model be built that will produce an optimized public transport network for South African metropolitan areas, taking the country's particulars into account?", is provided by proposing and developing a model consisting of four components: the Geographic Information System (GIS), the Population Generation Algorithm (PGA), the Bus Stop Placement Algorithm (BSPA), and the Multimodal Network Design Algorithm (MNDA). Each component makes its own contribution to the overall benefit of the model.

The GIS component provides the ability to organize the geographic data structurally through the use of a geodatabase. Through the design of the geodatabase, data validation is ensured by the addition of a *topology*, and the possibility of generating Origin-Destination (OD) distances using the actual network is ensured by the addition of a *network dataset*. The GIS also provides powerful visualization capabilities, enabling the presentation of the results of each of the other components of the model. The data flow between the GIS component and the three algorithmic components is ensured through transformation of the data into a shape file format—the format that can be read by all the components. Two versions of the designs of the GIS are proposed: one that supports the design of the PGA used in this project, and one that supports a future design of the PGA, based on survey, census, and land use data. The current design of the PGA provides good results for the purposes of this project.

The BSPA component places bus stops in the study according to geographic input data—a new approach not found in the literature. The search of the Simulated Annealing (SA) based metaheuristic is improved by recalculating the perturbation weights every predetermined number of iterations, enabling the algorithm to adopt to its position in the solution space. The bus stops of the optimized solution are compared with the actual road network in order to produce a realistic solution. The algorithm provides the ability to specify certain bus stops as fixed, creating a more flexible algorithm that can be used to model a network that includes a train line or a fixed bus route. The BSPA provides freedom to the modeller—on the choice of the eventual network—by allowing him to choose the values for the parameters of the objective function weights, the maximum walking distance to a bus stop, and the unmet demand penalty. The performance test of the algorithm on the test area showed that, with the correct parameter combination, a good result can be generated, which is repeatable with a Mean Absolute Deviation (MAD) value of less than 1%. By applying the algorithm to the whole area of the City of Tshwane Metropolitan Municipality (CTMM), the BSPA is shown to be capable of optimizing real world and large scale networks.

The MNDA component enables the design of multimodal networks. The addition of the multimodal, shortest travel time, and asymmetric demand components to the solution approach proposed by Fan and Machemehl (2006b), creates an algorithm that simulates the real world more accurately. The time needed for the SA-based metaheuristic to find good solutions is reduced by basing the probability of choosing routes on the cost-per-utilization of the routes in the current solution. The vehicle change procedure was improved by removing the random choice component, and replacing it with a procedure that changes only the routes with the maximum or minimum vehicle frequencies. The performance test of the MNDA on a test network showed that the algorithm is capable of finding good solutions. With the appropriate parameters combination, a MAD value of less than 1% can be achieved. The results of the MNDA, applied to the CTMM, show that the algorithm is capable of designing real world and large-scale multimodal networks, although the algorithm needs significant computational resources.

The main addition to the body of knowledge resulting from this project is the optimal placement of bus stops according to geographic data of commuters, and the ability of the model to propose a network design for multiple vehicle types and multiple public transport modes.

7.2 Recommendations for future research

It is assumed that the development of a PGA component that generates commuter points based on survey, census, and land use data will produce a more realistically simulated population. The necessary geographic input data needed to develop this new component is available, and the matching geodatabase design is already presented in this dissertation. The improvement of the model has recently been initiated by the research group by assigning a Master's student who will focus on developing a new demand generation model based on recent and accurate geographic data, such as survey, census, and land use data.

In an attempt to model the behaviour of commuters more accurately, the option for commuters to walk to another bus stop—instead of always choosing the closest—could be added to the MNDA. In such a case the BSPA's list specifying which commuter is connected to which bus stop should be extended to multiple lists that contain, for example, the four closest bus stops for each commuter. The down-side of this addition is that it further increases of the computational resources needed to run the MNDA.

In order to reduce the computational resources needed to run the algorithms—and in particular the MNDA—the code of the algorithms could be re-evaluated and their efficiency improved.

A different approach to the alignment of the bus stops to the actual road network could be followed for the BSPA. Instead of performing the alignment in the last step of the algorithm, a solution space could be created that only includes positions that align with the road network. The MOVE STOP and INCREASE STOPS procedures would then need to be modified, so that new coordinates are created by selecting a random line from the road network on which a random point is created.

As part of a larger research project the model proposed in this dissertation will be used as a first component of a tool that will assist transport planners to evaluate changes to transport networks. The second component consists of a Multi Agent Transport Simulation (MATSim) toolkit that can evaluate the effect of changes to the transport network—for example, the adjustment of a current public transport network according to the results of the model proposed in this dissertation. The larger research project initially focuses on South African metropolitan areas, but has the potential to be used for similar transport networks in other developing and emerging economies.

Bibliography

Alterkawi, M. M. (2006). A computer simulation analysis for optimizing bus stops spacing: A case of riyadh, saudi arabia. *Habitat International*, 30(3):500–508.

Arctur, D. and Zeiler, M. (2004). *Designing geodatabses: Case studies in GIS data modeling*. ESRI, Inc.

Ceder, A. and Wilson, N. H. M. (1986). Bus network design. *Transportation Research*, 20B(4):331–344.

Chakroborty, P. (2003). Genetic algorithms for optimal urban transit network design. *Computer-Aided Civil and Infrastructure Engineering*, 18(3):184–200.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Fan, W. (2004). Optimal transit route network design problem: Algorithms, implementations, and numerical results. Phd thesis, University of Texas.

Fan, W. and Machemehl, R. B. (2004). A tabu search based heuristic method for the transit route network design problem. In *The 9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT)*, San Diego.

Fan, W. and Machemehl, R. B. (2006a). Optimal transit route network design problem with variable transit demand: Genetic algorithm approach. *Journal of Transportation Engineering*, 132(1):40–51.

Fan, W. and Machemehl, R. B. (2006b). Using a simulated annealing algorithm to solve the transit route network design problem. *Journal of Transportation Engineering*, 132(2):122–132.

Freeman, M. A. and Minkwitz, R. (2002). Applied optimization using matlab. Online webinar recorded on 4 December 2002.

Hao, Q., Soong, B., Gunawan, E., Ong, J., Soh, C., and Li, Z. (1997). A low-cost cellular mobile communication system: A hierarchical optimization network resource planning approach. *IEEE Journal on Selected Areas in communications*, 15(7):1315–1326.

Jha, M. K., McCall, C., and Schonfeld, P. (2001). Using GIS, genetic algorithms, and visualization in highway development. *Computer-Aided Civil and Infrastructure Engineering*, 16(6):399–414.

Joburg (2007). Tshwane income per employed person. Available online at www.joburg.org.za (retrieved on 3 September 2007).

Kuan, S. N., Ong, H. L., and Ng, K. M. (2004). Applying metaheuristics to feeder bus network design problem. *Asian-Pacific Journal of Operational Research*, 21(4):543–560.

Lampkin, W. and Saalmans, P. (1967). The design of routes, service frequencies and schedules for a municipal bus undertaking: A case study. *Operational Research Quarterly*, 18(4):375–397.

Lee, Y. and Vuchic, V. R. (2005). Transit network design with variable demand. *Journal of Transportation Engineering*, 131(1):1–10.

Li, X. and Yeh, A. G. O. (2005). Integration of genetic algorithms and GIS for optimal location search. *International Journal of Geographical Information Science*, 19(5):581–601.

Murray, A. T. (2001). Strategic analysis of public transport coverage. *Socio-Economic Planning Sciences*, 35(3):175–188.

Ngamchai, S. and Lovell, D. R. (2003). Optimal time transfer in bus transit route network design using a genetic algorithm. *Journal of Transportation Engineering*, 129(5):510–521.

Pattnaik, S. B., Mohan, S., and Tom, V. M. (1998). Urban bus transit route network design using genetic algorithm. *Journal of Transportation Engineering*, 124(4):368–375.

Rardin, R. L. (1998). *Optimization in Operations Research*. Prentice Hall.

Saka, A. A. (2001). Model for determining optimum bus-stop spacing in urban areas. *Journal of Transportation Engineering*, 127(3):195–199.

Schöbel, A. (2005). Locating stops along bus and railway lines —a bicriteria problem. *Annals of Operations Research*, 136(1):211–227.

Shrivastava, P. and Dhingra, S. L. (2001). Development of feeder routes for suburban railway stations using heuristic approach. *Journal of Transportation Engineering*, 127(4):334–341.

Shrivastava, P. and Dhingra, S. L. (2002). Development of coordinated schedules using genetic algorithms. *Journal of Transportation Engineering*, 128(1):89–96.

Shrivastava, P. and O'Mahony, M. (2006). A model for development of optimized feeder routes and coordinated schedules —a genetic algorithms approach. *Transport Policy*, 13(5):413425.

Tat, C. W. and Tao, F. (2003). Using GIS and genetic algorithm in highway alignment optimization. *Intelligent Transportation Systems*, 2(2):1563–1567.

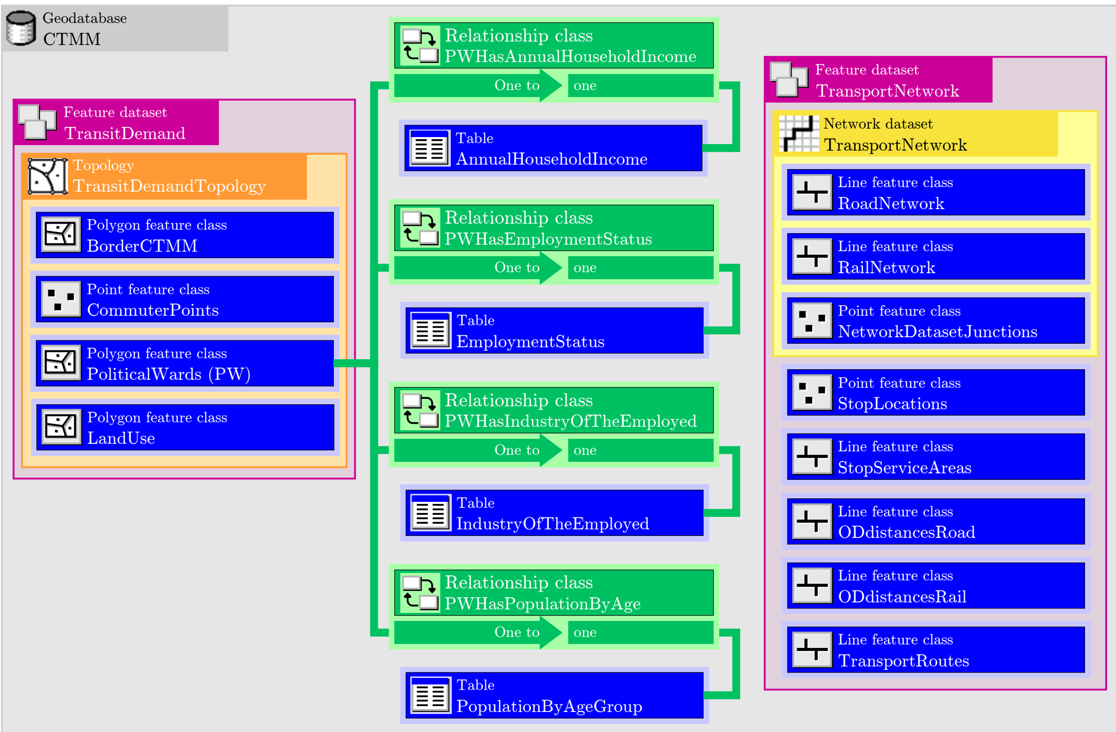
Tom, V. M. and Mohan, S. (2003). Transit route network design using frequency coded genetic algorithm. *Journal of Transportation Engineering*, 129(2):186–195.

Yen, J. Y. (1971). Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716.

Zhao, F. (2006). Large-scale transit optimization by minimizing user cost and transfers. *Journal of Public Transportation*, 9(2):107–129.

Appendix A

Alternative design of the GIS component



Appendix B

Detailed description of the elements in the geodatabase

Simple feature class
BorderCTMM

Geometry *Polygon*
Contains M values *No*
Contains Z values *No*

Metropolitan boundary of CTMM

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length
OBJECTID	Object ID						
Shape	Geometry	Yes					
Shape_Length	Double	Yes			0	0	
Shape_Area	Double	Yes			0	0	

Simple feature class
CommuterPoints

Geometry *Point*
Contains M values *No*
Contains Z values *No*

Position of the commuter points in the CTMM

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length	
OBJECTID	Object ID							
Shape	Geometry	Yes						
CommuterID	Long integer	Yes			0			ID of the commuter point
Type	String	Yes					10	Production or attraction point
Link	Long integer	Yes			0			Links each production point to an attraction point

Simple feature class
TransportationZones

Geometry *Polygon*
Contains M values *No*
Contains Z values *No*

Transportation zones of the CTMM

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length	
OBJECTID	Object ID							
Shape	Geometry	Yes						
ID	String	Yes					50	ID of the transportation zones
Shape_Length	Double	Yes			0	0		
Shape_Area	Double	Yes			0	0		

Simple feature class
ODdistancesRail

Geometry *Polyline*
Contains M values *No*
Contains Z values *No*

Distances between the train stations

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length	
OBJECTID_1	Object ID							ID of OD line
Shape	Geometry	Yes						
ObjectID	Long integer	Yes			0			
Name	String	Yes					128	Describes the origin and destination stations
OriginID	Long integer	Yes			0			ID of the origin station
Destinatio	Long integer	Yes			0			ID of the destination station
Total_Mete	Double	Yes			0	0		Distance between the origin and destination station
Shape_Length	Double	Yes			0	0		

Simple feature class
ODdistancesRoad

Geometry *Polyline*
Contains M values *No*
Contains Z values *No*

Distances between the bus stops

Field name	Data type	Allow nulls	Default value	Domain	Precision	Scale	Length	
OBJECTID_1	Object ID							ID of OD line
Shape	Geometry	Yes						
ObjectID	Long integer	Yes			0			
Name	String	Yes					128	Describes the origin and destination bus stops
OriginID	Long integer	Yes			0			ID of the origin bus stop
Destinatio	Long integer	Yes			0			ID of the destination bus stop
Total_Mete	Double	Yes			0	0		Distance between the origin and destination bus stop
Shape_Length	Double	Yes			0	0		

Simple feature class
RailNetwork

Geometry *Polyline*
Contains M values *No*
Contains Z values *No*

Rail line of the CTMM

Field name	Data type	Allow nulls	Default value	Domain	Precision	Scale	Length	
OBJECTID	Object ID							
Shape	Geometry	Yes						
TYPE	String	Yes					100	Type of rail line
CITYNAME	String	Yes					30	Name of the City containing the rail line
PROVINCE	String	Yes					30	Name of the Province containing the rail line
VERSION	String	Yes					12	Indicating the year and the month of release
Shape_Length	Double	Yes			0	0		

Simple feature class
RoadNetwork

Geometry *Polyline*
Contains M values *No*
Contains Z values *No*

Road network of the CTMM

Field name	Data type	Allow nulls	Default value	Domain	Precision	Scale	Length	
OBJECTID	Object ID							
Shape	Geometry	Yes						
PREFIX	String	Yes					2	Directional prefix of the street name
NAME	String	Yes					50	Name of the road
TYPE	String	Yes					15	Description of the road
SUFFIX	String	Yes					2	Directional Suffix of the street name
FULL_NAME	String	Yes					50	Complete street name
ALT_NAME	String	Yes					50	Alternate street name
LABEL	String	Yes					50	Used for labeling maps
ROUTE	String	Yes					30	The route number of the road
RDCATEGORY	String	Yes					50	Category of the road
PROVINCE	String	Yes					30	Name of the Province containing the road
MUNICIPAL	String	Yes					50	Name of the Municipality containing the road
CITYNAME	String	Yes					30	Name of the City containing the road
OLDCITY	String	Yes					35	Old City name
SUB_LEFT	String	Yes					35	Suburb on the left side of the road
SUB_RIGHT	String	Yes					35	Suburb on the right side of the road
F_ZLEV	Long integer	Yes			0			Elevation level of FROM node
T_ZLEV	Long integer	Yes			0			Elevation level of TO node
ONEWAY	String	Yes					2	Direction of travel on the road
RAMP_FLAG	String	Yes					7	Indicates if a road is an off-ramp or on-ramp
BR_FLAG	String	Yes					1	Indicates if road is a bridge link
TN_FLAG	String	Yes					1	Indicates if road is a tunnel link
METERS	Double	Yes			0	0		Length of the road in meters
SPEED_LIM	Long integer	Yes			0			Maximum allowed travel speed
L_F_ADD	Long integer	Yes			0			Beginning address range on left side of the road
L_T_ADD	Long integer	Yes			0			Ending address range on left side of the road
R_F_ADD	Long integer	Yes			0			Beginning address range on right side of the road
R_T_ADD	Long integer	Yes			0			Ending address range on right side of the road
ID	Long integer	Yes			0			ID of the road
VERSION	String	Yes					12	Indicating the year and the month of release
Shape_Length	Double	Yes			0	0		

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Simple feature class
StopLocations

Geometry *Point*
Contains M values *No*
Contains Z values *No*

Positions of the stops

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length	
OBJECTID	Object ID							
Shape	Geometry	Yes						
StopNr	Short integer	Yes			0			ID of the stops
Status	Short integer	Yes			0			Indicating whether a stop has a fixed status

Simple feature class
StopServiceAreas

Geometry *Polyline*
Contains M values *No*
Contains Z values *No*

Service areas of the stops

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length	
OBJECTID	Object ID							
Shape	Geometry	Yes						
StopNr	Short integer	Yes			0			ID of the stops
Status	Short integer	Yes			0			Indicating whether a stop has a fixed status
Shape_Length	Double	Yes			0	0		

Simple feature class
NetworkDatasetJunctions

Geometry *Point*
Contains M values *No*
Contains Z values *No*

Junctions of the network dataset

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length	
OBJECTID	Object ID							
SHAPE	Geometry	Yes						
ZELEV	Long integer	Yes			0			Indicating elevation level of junction

Simple feature class
TransportRoutes

Geometry *Polyline*
Contains M values *No*
Contains Z values *No*

Transport routes

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length	
OBJECTID	Object ID							
Shape	Geometry	Yes						
Line	Short integer	Yes			0			ID of the routes
Type	Short integer	Yes			0			Vehicle type operating on the route
Number	Short integer	Yes			0			Number of vehicles operating on the route
Shape_Length	Double	Yes			0	0		

Table
Emme2DemandData

Demand data from the EMME/2 model

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length	
OBJECTID	Object ID							
ORIGIN_TZ	String	Yes					9	Origin transportation zones
DEST_TZ_1	String	Yes					9	Demand to destination transportation zone 1
___	String	Yes					9	Demand to destination transportation zone 2 till N-1
DEST_TZ_N	String	Yes					9	Demand to destination transportation zone N

Relationship class
TransportationZoneHasDemand

Relation between transportation zones and the demand data

Type *Simple*
Cardinality *One to one*
Notification *None*

Forward label *Emme2DemandData*
Backward label *TransportationZones*

Origin feature class	Destination table
Name *TransportationZones* Primary key *ID* Foreign key *ORIGIN_TZ*	Name *Emme2DemandData*

No relationship rules defined.

Appendix C

Parmeter values used to determine baseline for sensitivity anlysis on test area BSPA

parameter combination	tempReductionFreq	tempReductionFactor	tempObjectiveFactor	minTemp	initialNrStops	maxPossibleMoves	perburbationFreq
1	25	0.85	1	50	10	5	1
2	5	0.85	0.05	100	30	5	50
3	35	0.8	0.05	5	10	20	50
4	5	0.75	0.1	50	10	100	10
5	25	0.9	5	50	10	50	1
6	45	0.85	0.05	50	10	50	1
7	5	0.85	5	100	20	10	10
8	35	0.75	0.05	100	40	50	5
9	45	0.95	1	50	20	5	1
10	15	0.75	1	50	40	10	10
11	45	0.75	0.1	100	10	50	10
12	15	0.75	0.05	5	50	100	100
13	5	0.85	0.1	5	30	5	100
14	45	0.9	5	100	10	5	10
15	25	0.95	5	50	30	50	100
16	25	0.8	0.1	50	20	5	10
17	35	0.85	0.1	1	10	20	50
18	25	0.75	5	100	50	50	5
19	25	0.95	1	1	10	50	1
20	25	0.75	0.05	10	50	5	10
21	35	0.8	5	100	50	5	1

continued on next page

continued from previous page

22	35	0.9	5	100	50	10	50
23	5	0.75	0.1	10	50	5	100
24	45	0.8	5	50	20	20	10
25	35	0.75	0.5	1	20	50	5
26	15	0.95	0.1	100	20	50	50
27	15	0.95	1	50	20	10	50
28	5	0.95	1	5	30	20	1
29	45	0.85	0.1	100	10	20	5
30	**15**	**0.95**	**0.05**	**5**	**20**	**100**	**100**
31	35	0.75	5	100	40	10	100
32	25	0.8	0.05	100	10	100	1
33	45	0.75	0.05	1	40	5	100
34	5	0.8	0.05	10	30	100	1
35	35	0.75	0.5	100	20	20	1
36	15	0.75	0.1	100	10	100	100
37	5	0.8	0.05	50	10	5	10
38	45	0.95	0.5	10	50	20	10
39	5	0.95	5	100	40	20	5
40	25	0.8	5	100	50	100	5
41	25	0.95	0.5	10	10	5	1
42	25	0.9	0.1	10	10	10	10
43	15	0.95	0.5	5	40	100	50
44	45	0.8	0.05	10	50	5	10
45	5	0.85	0.5	100	10	100	1
46	25	0.95	5	100	20	10	5
47	25	0.85	0.5	100	50	50	1
48	45	0.75	0.1	100	30	20	5
49	45	0.75	0.1	5	10	5	5
50	15	0.8	0.1	10	40	100	100

Appendix D

Input data for test network MNDA

Table D.1: X- and Y-coordinates of stops in the test network of the MNDA

Stop	X-coordinate	Y-coordinate
1	13.4	2.1
2	13.4	0.6
3	11.8	1.2
4	11.8	3.7
5	11.8	4.8
6	11.8	7.3
7	13.4	7.9
8	11.8	8.1
9	9.8	1.8
10	9.0	2.0
11	7.7	4.7
12	7.3	10.3
13	7.5	0.3
14	7.7	2.3
15	5.8	0.0
16	5.6	2.9
17	5.8	6.5
18	5.8	10.3
19	4.4	10.3
20	4.4	6.0
21	4.0	3.6
22	1.6	5.1
23	2.9	10.3
24	1.2	10.3
25	0.0	6.1

Table D.2: Demand matrix of the test network of the MNDA

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
1	-	100	150	150	10	20	15	125	115	50	150	115	50	21	115	60	100	60	50	100	35	10	10	10	10
2	100	-	10	2	2	1	3	3	4	5	3	3	3	1	3	3	3	3	2	5	4	2	1	1	1
3	150	10	-	150	10	150	250	150	50	150	25	10	115	25	10	10	15	5	5	150	25	10	5	50	5
4	150	2	150	-	25	25	25	10	150	100	50	30	20	35	7	80	12	25	30	20	30	10	20	50	5
5	10	2	10	25	-	30	25	10	15	10	10	15	25	40	45	30	10	15	10	25	10	10	25	50	20
6	20	1	150	25	30	-	15	30	10	15	5	10	10	15	25	30	25	20	30	30	20	10	25	100	20
7	15	3	250	25	25	15	-	2	2	2	1	2	3	5	4	3	3	1	1	3	2	1	3	5	3
8	125	3	150	10	10	30	2	-	30	30	25	10	10	10	10	10	15	50	25	25	5	10	40	75	25
9	115	4	50	150	15	10	2	30	-	10	10	5	7	7	15	15	15	15	15	50	5	5	10	65	25
10	50	5	150	100	10	15	2	30	10	-	3	3	3	3	3	3	3	3	3	2	5	10	10	100	25
11	150	3	25	50	10	5	1	25	10	3	-	4	4	4	4	4	2	2	3	5	5	5	5	35	25
12	115	3	10	30	15	10	2	10	5	3	4	-	40	40	40	40	40	40	40	50	5	10	5	40	25
13	50	3	115	20	25	10	3	10	7	3	4	40	-	35	25	30	25	20	20	50	5	15	15	25	25
14	21	1	25	35	40	15	5	10	7	3	4	40	35	-	40	40	35	30	30	10	5	20	10	25	25
15	115	3	10	7	45	25	4	10	15	3	4	40	25	40	-	45	100	50	50	15	5	35	25	25	25
16	60	3	10	80	30	30	3	10	15	3	4	40	30	40	45	-	10	25	25	20	5	40	25	25	20
17	100	3	15	12	10	25	3	15	15	3	2	40	25	35	100	10	-	10	5	100	10	40	30	25	40
18	60	3	5	25	15	20	1	50	15	3	2	40	20	30	50	25	10	-	10	100	10	40	20	25	40
19	50	2	5	30	10	30	1	25	15	3	3	40	20	30	50	25	5	10	-	100	15	25	5	5	40
20	100	5	150	20	25	30	3	25	50	2	5	50	50	10	15	20	100	100	100	-	10	25	10	5	40
21	35	4	25	30	10	20	2	5	5	5	5	5	5	5	5	5	10	10	15	10	-	10	10	5	40
22	10	2	10	10	10	10	1	10	5	10	5	10	15	20	35	40	40	40	25	25	10	-	10	5	40
23	10	1	5	20	25	25	3	40	10	10	5	5	15	10	25	25	30	20	5	10	10	10	-	5	40
24	10	1	50	50	50	100	5	75	65	100	35	40	25	25	25	25	25	25	5	5	5	5	5	-	40
25	10	1	5	5	20	20	3	25	25	25	25	25	25	25	25	20	40	40	40	40	40	40	40	40	-

Appendix E

Parmeter values used to determine baseline for sensitivity anlysis on test network MNDA

parameter combination	tempReductionFreq	tempReductionFactor	tempObjectiveFactor	minTemp	routeChangeProbability	localMoveProbability	numberOfRoutes
1	25	0.85	0.05	50	0.9	0.7	7
2	5	0.8	0.5	5	0.3	0.7	13
3	35	0.75	0.5	100	0.1	0.1	13
4	25	0.9	0.1	10	0.1	0.3	7
5	45	0.8	0.5	10	0.1	0.3	7
6	5	0.85	0.1	5	0.1	0.9	11
7	**25**	**0.95**	**0.5**	**5**	**0.7**	**0.3**	**9**
8	45	0.95	0.05	10	0.1	0.7	7
9	15	0.75	0.05	10	0.5	0.9	11
10	35	0.95	1	5	0.1	0.3	11
11	5	0.95	0.5	100	0.9	0.7	5
12	5	0.8	1	100	0.3	0.9	5
13	45	0.9	0.1	10	0.9	0.7	11
14	25	0.95	0.1	10	0.5	0.5	5
15	25	0.75	1	10	0.1	0.7	11
16	35	0.8	1	1000	0.1	0.1	13
17	25	0.75	0.1	5	0.9	0.3	9
18	25	0.95	0.05	1000	0.1	0.3	7
19	15	0.95	0.5	50	0.7	0.7	11
20	35	0.8	0.1	5	0.7	0.7	7
21	35	0.9	0.1	5	0.7	0.9	13

continued on next page

continued from previous page

22	45	0.8	0.1	10	0.3	0.1	11
23	25	0.95	5	1000	0.3	0.3	9
24	15	0.9	1	5	0.3	0.3	13
25	15	0.95	0.05	10	0.1	0.9	13
26	5	0.95	0.05	100	0.5	0.1	7
27	45	0.8	1	5	0.1	0.1	9
28	15	0.9	0.5	100	0.3	0.7	5
29	35	0.75	0.1	5	0.7	0.1	5
30	25	0.75	0.5	5	0.1	0.5	7
31	35	0.95	0.5	1000	0.5	0.9	5
32	5	0.75	0.5	50	0.5	0.5	7
33	25	0.95	5	5	0.3	0.1	7
34	5	0.95	1	5	0.1	0.7	5
35	5	0.75	0.5	50	0.9	0.7	11
36	45	0.9	5	50	0.9	0.1	11
37	5	0.95	0.1	5	0.7	0.1	9
38	25	0.8	0.1	5	0.9	0.5	9
39	25	0.9	5	100	0.9	0.7	7
40	25	0.85	1	100	0.9	0.9	11
41	15	0.9	5	100	0.7	0.5	13
42	45	0.75	0.5	50	0.7	0.7	11
43	5	0.8	5	5	0.1	0.5	7
44	25	0.95	0.1	5	0.1	0.9	9
45	25	0.8	5	5	0.9	0.3	7
46	35	0.95	1	5	0.5	0.1	9
47	35	0.95	1	1000	0.9	0.7	9
48	15	0.75	1	50	0.7	0.7	5
49	45	0.8	0.5	50	0.9	0.1	5
50	35	0.9	0.1	100	0.9	0.7	9

Figure 4.8. The graph shows the typical characteristics of an SA metaheuristic: initially the algorithm searches widely through the solution space to ensure divergence, and as the temperature decreases in subsequent iterations the solution slowly converges to the best local optimum found. It seems, because of the scale of the graph, that after 11 000 iterations no improvements are made. However, this is not true: small improvements are still made until iteration 17 089, when the final incumbent value is found.

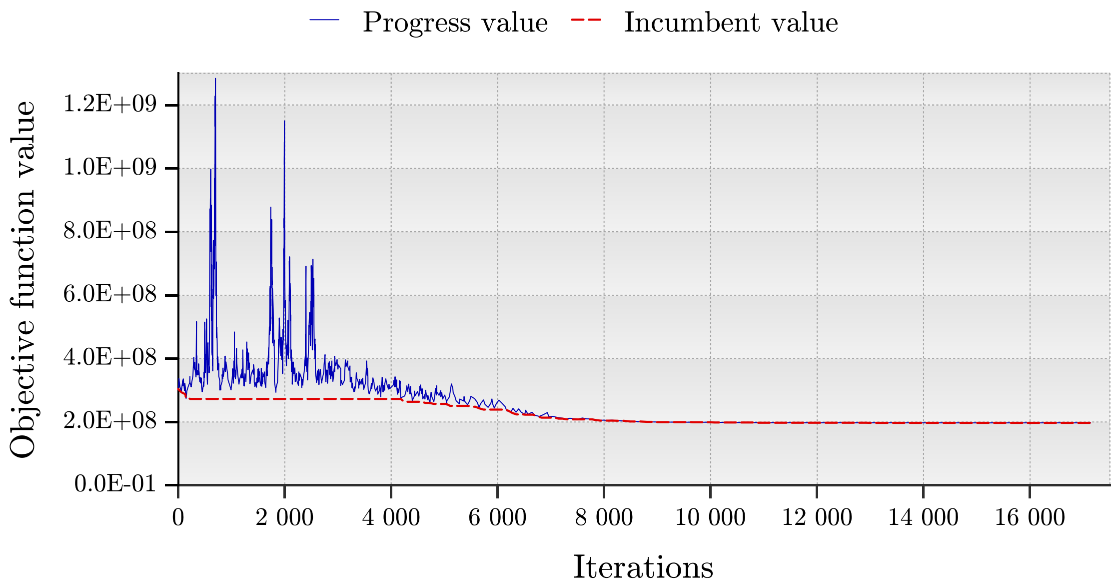


Figure 4.8: Progress of the algorithm during one run

4.4 Conclusion

In this chapter the development of the BSPA has been presented. The proposed model tries to find an optimal solution between the commuters' walking distance to a bus stop and the number of bus stops placed in the study area. An SA metaheuristic has been developed that applies a change to the current solution by adding, deleting, or moving a bus stop from the current solution. To increase the search intelligence of the algorithm, the probability of applying one of these changes is recalculated after a predetermined number of iterations. After the completion of the search for a better solution, the bus stops are placed on realistic points in the study area by comparing their current position with the road network of the study area.

The algorithm has been tested on a test area, for which the appropriate weights for the two components of the objective function were determined. The sensitivity analysis showed the effect of the parameter values on the objective function value and provided a combination of parameter values that produces good and repeatable solutions. The proposed solution was presented, showing that the algorithm is capable of finding good solutions to the bus stop placement problem.

The results from the BSPA are used as input data for the next component of the main model. The development of this component is presented in the next chapter.