APPLYING METAHEURISTICS TO FEEDER BUS NETWORK DESIGN PROBLEM

KUAN SZE NEE

NATIONAL UNIVERSITY OF SINGAPORE

2003

APPLYING METAHEURISTICS TO FEEDER BUS NETWORK DESIGN PROBLEM

KUAN SZE NEE
(B.Eng. (Hons.). NUS)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2003

Acknowledgements

I would like to sincerely express my gratitude to my research supervisor, A/P Ong Hoon Liong, for his guidance throughout the course of my research in the past two years. He has generously imparted his valuable knowledge of optimization and programming techniques to me and has patiently guided me to competency. Without his support and encouragement, I would not been able to achieve this far.

I would also like to thank the following people who have helped me a lot in various ways to the success completion of this thesis: Teng Suyan, Liu Shubin, Zhang Caiwen, Tan Yan Ping, Cheong Wee Tat, all students in Ergonomics Lab, and all friends in the Industrial and Systems Engineering Department in NUS.

Lastly, I would also like to thank my family and friends for their encouragement and support to complete this degree. I have certainly gained a lot of knowledge from this course and acquainted with like-minded friends whom we shared life experience with.

Table of contents

Summary

Route network design is the first and the most important step in the bus transportation planning process. This is because the route structure designed becomes an important input to the subsequent decision making processes and will invariably affect the later planning steps. Effective design of route network and service frequencies can decrease the overall cost of providing the transit service and increase the efficiency of the bus transit system. The main challenge of the route network design problem is to be able to give a good and efficient solution in a reasonable computation time.

In this thesis, we focus on the design of the Feeder Bus Network Design Problem (FBNDP). The problem involves designing a set of feeder bus routes to provide access to an existing rail public transport system and the determination of the operating frequency on each route, such that the objective function of the total operator and user costs is minimized. The main objective of this research is to design better and more efficient algorithms to the FBNDP by exploring the use of metaheuristics and other innovative heuristics. Metaheuristics have their own way of avoiding getting trapped in local minimum in its search for the global minimum and have seldom being used to solve the FBNDP in the literature. Their potential is explored in this research. Four metaheuristics for solving the FBNDP are proposed. They are Simulated Annealing, Tabu Search, Genetic Algorithm and the more recent metaheuristic, Ant Colony Optimization. The results are compared to those published in literature. A comparative study is also carried out on several test problems generated at random to evaluate the performance of these heuristics in terms of their computational efficiency and solution quality. These problems vary in several

characteristics such as the problem size and the problem structure. The problem size is determined by the size of the service area, the number of stations in the service area and the density of the bus stops in the service area. The problem structure is determined by the shape of the station network: a line network, a junction network and a crossing network, the location of the destination station: either at the central or at the peripheral of the service area, and the grouping of the bus stops: either clustered or evenly distributed based on the geographical location of the service area.

Computational experiments have shown that Tabu Search combined with an intensification strategy is the most effective metaheuristic, generating better quality solutions. It has also produced a new best solution as compared to the literature. However, the computational time is the longest. Genetic Algorithm is closely comparable to basic Tabu Search. Simulated Annealing is fast and offers reasonably good solutions. Ant Colony Optimization is comparable to the state-of-the-art algorithms such as Simulated Annealing.

Nomenclature

ACO	Ant colony optimization
CBD	Central Business District
FBNDP	Feeder Bus Network Design Problem
GA	Genetic algorithm
MDVRP	Multi-depot vehicle routing problem
SA	Simulated annealing
TS	Tabu search
TSP	Travelling salesman problem
VRP	Vehicle routing problem

cooling_rate	Constant for the reduction of the temperature for SA
max_count	Maximum counter value for stopping criterion for SA
max_Iter	Maximum number of iterations for stopping criterion for TS and ACO
max_moves	Maximum number of moves to be performed at each temperature *T* for SA
min_percent	Minimum percentage to define thermal equilibrium at each temperature for SA
n_gen	Number of generations for GA
n_NB	Size of nearest neighbours for TS and ACO
pop_size	Size of population for GA
tabu_size	Tabu list size for TS

ASH	Maximum available seat-hours
C_{js}	Unit rail wait-time and riding-time cost from rail node j to destination s ($/passenger)
c	Bus-vehicle capacity
D_E	Exchange delimiter

D_I	Initial delimiter
D_k	Maximum route length of route *k* (miles)
D_{ij}	Distance between stops *i* and *j*
d_{ij}	Distance between stops *i* and *j* for ACO
d_i^{st}	Distance between stop *i* and station *st* for ACO
$\widetilde{d}_i^{st}$	Modified distance from station *st* to stop *i* for ACO
F_k	Service frequency on bus route *k*
H	Any proper subset of *N* containing the set of all rail nodes
I	Number of stops
$INCOST_i(p,q)$	Cost of inserting stop *i* between nodes *p* and *q* of route *k*
J	Number of stations
K	Number of routes
L_{ih}	Distance between nodes *i* and *h* (miles)
L_k	Length of route segment *k*
l_{ij}	Distance from stop *i* to station *j*
$MSVTC_i(p,q)$	Modified savings from inserting stop *i* between nodes *p* and *q* of route *k*
MTC_i^j	Modified cost of direct route from stop *i* to station *j*
m	Number of ants for ACO
N	Set of all nodes
P_c	Crossover probability for GA
P_m	Mutation probability for GA
p_{ij}^k	Probability of ant *k* of visiting stop *j* from stop *i* for ACO
$p_{i,st}^k$	Probability of ant *k* choosing to link stop *i* to station *st* for ACO
Q	Quantity of pheromone laid by an ant per iteration for ACO
Q_i	Average demand per hour at bus node *i* (passengers/hour)
Q^k	Demand on route segment *k*

$\overline{Q}$	Average hourly demand per stop (passengers/hour)
q_i	Average demand per hour at bus stop i
$SAVTC_i^k$	Savings from including stop i in route segment k
s_{ij}^{st}	Savings from linking stops i and j to a route assigned to station st for ACO
$\widetilde{s}_{ij}^{st}$	Modified savings from linking stops i and j to a route assigned to station st for ACO
T	Current temperature for SA
T_i	Temperature at iteration i for SA
T_{init}	Initial temperature for SA
TC_i^j	Total cost of direct route from stop i to station j
TC^{μ}	Total cost generated by the μth best ant for ACO
TC^*	Total cost of the best found solution for ACO
$TC(s)$	Total cost for solution s
U	Average bus operating-speed (mile/hour)
X_{ihk}	Binary variable; Value of 1 if stop i precedes stop h on bus route k
Y_{ij}	Binary variable; Value of 1 if stop i is assigned to station j
α	Parameter to regulate the influence of pheromone trail, τ_{ij} and $\tau_{i,st}$, for ACO
β	Parameter to regulate the influence of heuristic information, η_{ij} and $\eta_{i,st}$, for ACO
σ	Number of elitist ants for ACO
μ	Ranking index of ants for ACO
Ω	The set of stops which ant k has not yet visited for ACO
Ω_{st}	The set of stations which stop i can be assigned to for ACO
λ_0	Unit bus operating-cost ($/vehicle-mile)

λ_r	Value of one passenger-hour of riding-time ($/passenger-hour)
λ_w	Value of one passenger-hour of waiting-time ($/passenger-hour)
ρ_L	Average bus-load factor
ρ	Pheromone trail evaporation rate for ACO
η_{ij}	Visibility of stop j from stop i for ACO
$\eta_{i,st}$	Visibility of station st from stop i for ACO
τ_{ij}	Intensity of pheromone trail between stops i and j for ACO
$\tau_{i,st}$	Intensity of *pseudo*-pheromone trail between stop i and station st for ACO
τ_{ij}^{new}	Updated pheromone trail level on edge (i, j) for ACO
$\Delta\tau_{ij}^{*}$	Increase of trail level on edge (i, j) caused by the elitist ants for ACO
$\Delta\tau_{ij}^{\mu}$	Increase of trail level on edge (i, j) caused by the μth best ant for ACO

Subscripts:

h	Node index (bus node or rail node); $h = 1, \ldots, I + J$
i, m	Bus-node index; $i, m = 1, \ldots, I$
j	Rail-node index; $j = I + 1, \ldots, I + J$
k	Bus-route index; $k = 1, \ldots, K$
s	Destination station

List of Figures

List of Tables

Chapter 1 Introduction

In this chapter, we present the role of route network design in the transportation planning process and the difficulties faced by researchers in dealing with this problem. This research is focused on the feeder bus network design problem. The model formulation, the objective and the scope of this research are presented.

1.1 The Route Network Design Problem

The transportation planning process is decomposed into four basic elements performed in sequence due to its complexity: route network design and setting frequencies, timetabling, bus scheduling and driver scheduling. Route network design involves the design of an optimal network of bus routes and associated frequencies (inverse of headways) for each route that achieves some desired objectives, subject to some operational constraints and maintaining the quality of service offered. Being the first step in the planning process, it becomes the most important planning step in the bus transit planning process (Ceder and Wilson, 1986). This is because the route structure designed becomes an important input to the subsequent decision making processes and will invariably affect the later planning steps. It is also due to the fact that bus operators have the least flexibility in altering the routes once they are set.

Ceder and Wilson (1986) highlighted some important components of route network design such as estimating demand, identification of objective function, constraints and passenger behaviour. The objective functions adopted by various researchers vary widely. Ceder and Israeli (1998) explained that an ideal objective function should take care of the operator as well as the user perspectives in a transit operation. The prime concern of the operator is to minimize the operating cost or maximize the profit, which depends on the fleet size and the average bus round trip travel time and total distance traveled. The user, on the other hand, looks for minimum total travel time which may consist of waiting time, riding time and transfer time, the number of transfers and sometimes the access cost, which is incurred by passengers walking to the bus route. Hence, the objective function is generally the minimization of the sum of user costs and operator costs. Feasibility constraints may include minimum or maximum operating frequencies on all or selected routes, the maximum load factor on any bus route which is a function of vehicle size, the maximum allowable bus fleet size or operating costs, the vehicle size, etc.

In terms of passenger demand, most of the time it is assumed fixed or inelastic for simplicity. The assumption of fixed demand may be reasonable for systems in which ridership is insensitive or independent to service quality or price. Otherwise, demand can be a variable, probably due to the sharing or competition of the public transport. Two types of travel demand patterns, many-to-one and many-to-many, are also considered. The many-to-one demand pattern refers to passengers traveling from multiple origins to a single destination. This is usually more applicable to feeder bus services which carry passengers to a common destination (e.g. a central business district or a transfer station). In most bus services, the many-to-many demand pattern is considered when passengers have different origins and destinations.

Effective design of transit routes and service frequencies can decrease the overall cost of providing the transit service and increase the efficiency of the bus transit system, thereby attracting people to use the public transportation system. There are several criteria considered by researchers to evaluate the efficiency of bus route network design. In general, an efficient bus route network design should be able to satisfy most, if not all, of the existing transit demand, without requiring passengers to transfer from one route to another. The routes should be preferably short to reduce operating cost and the total travel time required by passengers to reach their destinations should be fast and passengers are able to easily access from their origins or destinations (e.g. a non-circuitous route). For a longer term planning process, a bus route design system should be more dynamic and can provide a quick solution to incident events such as accidents, road maintenance, or special events, without major disruptions to the present system. It should also be capable of designing the extension of existing bus routes or even redesigning the entire route network.

Another main challenge of the route network design problem is to be able to give a good and efficient solution in a reasonable computation time. As found out by some researchers, the problem of designing an efficient route network for a transit system is a difficult optimization problem which does not lend itself readily to mathematical programming formulations and solutions using traditional techniques (Newell, 1979). Similar observations are also made by Johnson et al. (1977) and Baaj and Mahmassani (1991). The transit route network design problem belongs to the class of NP-hard, combinatorial problems that suffer from several forms of mathematical complexity, such as nonlinearity, nonconvexity of the objective function, and the discrete and multiobjective nature of route design (Baaj and Mahmassani, 1990) which cannot be solved optimally by any polynomial growth algorithms (Johnson et al. 1977). This difficulty in solving the problem through traditional optimization techniques leads to the continual development of heuristic algorithms which are new and more robust to tackle such problems.

1.2 Types of Route Network Design Problems

There are two types of bus route network design problem. The first type is, given a service area with pre-specified bus stop locations and an hourly demand at each bus stop, the *bus network design problem* involves designing a set of bus routes and determining the associated frequencies for each route, such that it achieves the desired objective with a specified service level to the passengers and subject to some constraints imposed by the problem. In other words, the problem involves connecting all the demand points (bus stops) such that most, preferably all, the passengers are able to access from one point to another, while optimizing the objective function subject to the constraints imposed.

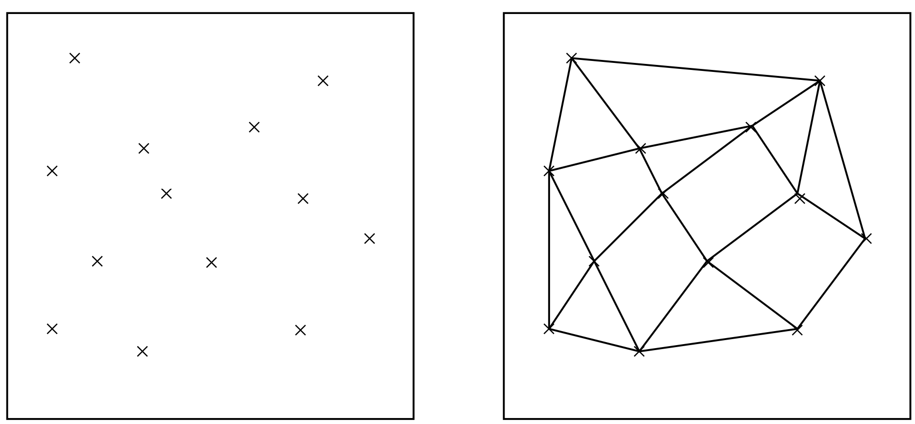

Figure 1.1: Bus network design problem

The second type is the FBNDP. It differs from the first type in that there is an existing rail public transport system (e.g. Mass Rapid Transit system in Singapore) and the buses serve to carry the passengers from the bus stops to the various stations. In other words, the problem involves designing a feeder bus network to provide access to an existing rail public transport system. Thus, given a service area with prespecified bus stop locations, and also a fixed rail transport system and an hourly demand at each bus stop, this problem also involves designing a set of feeder bus routes and determining the associated frequencies for each route that achieves the desired objective with a specified service level to the passengers and subject to the constraints.

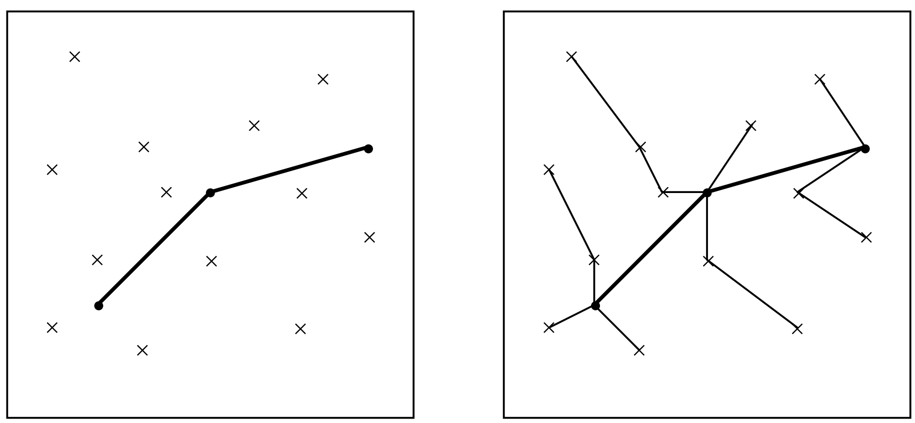

Figure 1.2: Feeder bus network design problem

An intermodal transit system, integrating a rail line and a number of feeder bus routes connected at different transfer stations, is inevitable in large metropolitan regions, where transit demand is high and widely spread. Singapore is a good example of such an integrated system. The present MRT system is fast, reliable, and highly efficient and is able to carry high volumes of passengers. It is a convenient mode of transport for the passengers and at the same time it eases the traffic on the roads. When this mode of transport exists, it is likely that a second level network of feeder bus routes are set up, whose main purpose is to transport users from the bus stops to the MRT network. The main challenge of designing an efficient feeder bus route network is to integrate and coordinate the rail and bus services as efficiently as possible. The development of better integrated intermodal systems improves service quality and passenger satisfaction that results from better coverage, reduced access costs, minimal delay and shorter travel times. From the viewpoint of the transit operators, an overall coordination among the various public transport modes can reduce their operating costs and increase their revenue by maintaining shorter routes and eliminating duplication of routes by the train and the buses. Thus, in this research we will focus on the feeder bus network design problem.

1.3 Problem Formulation

Notations

ASH	Maximum available seat-hours
C_{js}	Unit rail wait-time and riding-time cost from rail node j to destination s ($/passenger)
c	Bus-vehicle capacity
D_k	Maximum route length of route k (miles)
H	Any proper subset of N containing the set of all rail nodes
I	Number of stops
J	Number of stations
K	Number of routes
L_{ih}	Distance between nodes i and h (miles)
N	Set of all nodes

Q_i Average demand per hour at bus node i (passengers/hour)

$\overline{Q}$ Average hourly demand per stop (passengers/hour)

U Average bus operating-speed (mile/hour)

λ_0 Unit bus operating-cost ($/vehicle-mile)

λ_r Value of one passenger-hour of riding-time ($/passenger-hour)

λ_w Value of one passenger-hour of waiting-time ($/passenger-hour)

ρ_L Average bus-load factor

Subscripts:

h Node index (bus node or rail node); $h = 1, \ldots, I + J$

i, m Bus-node index; $i, m = 1, \ldots, I$

j Rail-node index; $j = I + 1, \ldots, I + J$

k Bus-route index; $k = 1, \ldots, K$

s Destination station

Then, the many-to-one (multiple origins, single destination) FBNDP can be formulated as a non-linear programming model with both continuous and integer variables as follows (Kuah and Perl, 1989):

Let

$$X_{ihk} = \begin{cases} 1 & \textit{if stop } i \textit{ precedes stop } h \textit{ on bus route } k \\ 0 & \textit{otherwise} \end{cases}$$

$$Y_{ij} = \begin{cases} 1 & \textit{if stop } i \textit{ is assigned to station } j \\ 0 & \textit{otherwise} \end{cases}$$

F_k = Service frequency on bus route k

Minimize $Z(X,Y,F)$

$$= \sum_{j=I+1}^{I+J} C_{js} \sum_{i=1}^{I} Q_i Y_{ij} + 2\lambda_0 \left[\sum_{k=1}^{K} F_k \sum_{i=1}^{I} \sum_{h=1}^{I+J} L_{ih} X_{ihk} \right]$$

$$+ \frac{\lambda_r}{2U} \sum_{k=1}^{K} \left(\sum_{i=1}^{I} \sum_{h=1}^{I+J} L_{ih} X_{ihk} \right) \left(\overline{Q} + \sum_{i=1}^{I} \sum_{h=1}^{I+J} Q_i X_{ihk} \right)$$

$$+ \lambda_w \left[\sum_{k=1}^{K} \frac{1}{2F_k} \sum_{i=1}^{I} \sum_{h=1}^{I+J} Q_i X_{ihk} \right] \quad (1.1)$$

subject to

$$\sum_{k=1}^{K} \sum_{h=1}^{I+J} X_{ihk} = 1; \qquad i = 1,\ldots,I \quad (1.1a)$$

$$\sum_{i=1}^{I} \sum_{j=I+1}^{I+J} X_{ijk} \le 1; \qquad k = 1,\ldots,K \quad (1.1b)$$

$$\sum_{h=1}^{I+J} X_{ihk} - \sum_{m=1}^{I} X_{mik} \ge 0; \qquad i = 1,\ldots,I\ ;\ k = 1,\ldots,K \quad (1.1c)$$

$$\sum_{i \notin H} \sum_{h \in H} \sum_{k=1}^{K} X_{ihk} \ge 1; \qquad \forall H \quad (1.1d)$$

$$\sum_{h=1}^{I+J} X_{ihk} + \sum_{m=1}^{I} X_{mjk} - Y_{ij} \le 1; \qquad i = 1,\ldots,I\ ;\ j = I+1,\ldots,I+J\ ;\ k = 1,\ldots,K \quad (1.1e)$$

$$\sum_{i=1}^{I} Q_i \sum_{h=1}^{I+J} X_{ihk} \le cF_k; \quad k = 1,\ldots,K \quad (1.1f)$$

$$\frac{2c}{\rho_L U} \sum_{k=1}^{K} F_k \sum_{i=1}^{I} \sum_{h=1}^{I+J} L_{ih} X_{ihk} \le ASH \quad (1.1g)$$

$$\sum_{i=1}^{I} \sum_{h=1}^{I+J} L_{ih} X_{ihk} \le D_k; \quad k = 1,\ldots,K \quad (1.1h)$$

$$X_{ihk},\ Y_{ij} = 0, 1; \quad i = 1,\ldots,I\ ;\ j = I+1,\ldots,I+J\ ;\ k = 1,\ldots,K\ ;\ h = 1,\ldots,I+J$$

$$F_k \ge 0; \quad k = 1,\ldots,K$$

The objective function consists of four cost components. The first term represents the passenger costs in the rail system (waiting-time and riding-time cost in the trains). The second term gives the operating costs incurred on the bus operator, which is proportional to the total distance traveled by the buses (round trip). The third term gives an approximation to the passengers' bus riding costs and the last term represents passengers' bus waiting costs. The constraints (1.1a) to (1.1e), together with the definition of the binary variables, determine the feasibility of the bus routes. Constraint (1.1a) requires that each stop is placed on a single route. Constraint (1.1b) ensures that each route is linked to a single station. Constraint (1.1c) constitutes the route-continuity, stating that a route which enters a bus node must leave that same node. Constraint (1.1d) requires that every feeder bus route must be linked to a station. Constraint (1.1e) specifies that a bus stop can be assigned to a station only if a route which terminates at that station passes through that stop. Constraints (1.1f), (1.1g) and (1.1h) impose the restrictions on the route network. Constraint (1.1f) specifies the limit on route capacity. Constraint (1.1g) states that the total seat-hours used should not exceed that which can be provided by the available fleet. Constraint (1.1h) imposes a maximum limit on each route length.

The FBNDP focuses on the design of a set of feeder bus routes and the determination of the operating frequency on each route, such that the objective function of the sum of operator and user costs is minimized. It can be viewed as achieving the optimal balance between the operator costs and user costs. The operator cost is related to the length travelled by the vehicles (total route length). The user cost is a function of total passenger travel time including the waiting time and riding time on both the bus and the train.

The network model, as shown in Figure 1.2, consists of two types of nodes – rail nodes and bus nodes, which represent railway stations and bus stops respectively. The rail transport network is assumed to be fixed, that is, defined in advance and not subjected to changes, and is represented as links joining the rail nodes shown in the diagram on the left in Figure 1.2. The location of bus stops is prespecified and the demand is assumed to be concentrated at bus nodes and inelastic to service quality and fares. The demand can be viewed as the average number of passengers per hour during the time period of study (e.g. peak period or off-peak period). The demand of passengers at each bus stop, the travel cost in the rail system between each pair of railway stations, the distance between each pair of bus stops and between every bus stop and every railway station are given. The bus fleet size, capacities and operating speed of the fleet of buses over the planning period are also given. In terms of network representation, the problem involves linking bus nodes to rail nodes, in which these bus links represent feeder bus route segments, as shown in the diagram on the right in Figure 1.2.

In our approach, FBNDP is considered under many-to-one demand pattern with multiple origins and a single destination. Peak-period work trips to the Central Business District (CBD) in the morning may exhibit this pattern. In this case, all passengers share a common destination, identified as the central city station. Passengers gathered at bus stops located in the service area wish to access this destination by first travelling by bus to any of the stations and then proceeding to the city centre by train. The following assumptions are made for the many-to-one FBNDP: (1) Each bus stop is served by one feeder bus route only. (2) Each feeder-bus route is linked to exactly one station. This implies that buses are not allowed to travel along rail lines, which is consistent with one of the basic purposes of integration, i.e. the elimination of duplicate services. (3) All buses have standard operating speeds and capacities. (4) The feeder-bus is assumed to halt at all the stops on its route.

1.4 Research Scope

A real world FBNDP is usually extremely large and complex. It cannot be solved to optimality within a reasonable amount of computation time using existing exact optimization methods and needs to be solved heuristically. The main objective of this research is to design efficient algorithms to the FBNDP by exploring the use of metaheuristics and other innovative heuristics. Metaheuristics have their own way of avoiding getting trapped in local minima in its search for the global minimum and have seldom been used to solve FBNDP. Their potential is explored in this research. Based on the formulation proposed in the literature, the model is solved using metaheuristics such as simulated annealing (SA), tabu search (TS) and genetic algorithms (GA), and a newer heuristic, the ant colony optimization (ACO). Results are compared to those published in literature for the benchmarking problem. A comparative study is also carried out on several test problems generated at random to compare the performance of the various heuristics in terms of better computational efficiency and better solution quality.

1.5 Organization of the Thesis

The thesis is arranged as follows. Chapter 2 presents a literature review of the published works that deal with bus network design problems and feeder bus network design problems. Chapter 3 describes our methodology for solving the FBNDP. Chapters 4, 5, 6 and 7 further describe the metaheuristics used and how we have applied them to solving the FBNDP. The overall computational results are summarized and analyzed in Chapter 8. Lastly, conclusions and future work are drawn in Chapter 9.

Chapter 2 Literature Review

In this chapter, we review the previous works related to the area of route network design. The literature is mainly classified into two groups: the bus network design problems (single-mode) and the feeder bus network design problems (multiple-mode with feeder buses serving a rail line or feeder buses serving the CBD). The different models and approaches to these problems are discussed.

2.1 Classification of Previous Approaches

A review of the literature reveals various approaches and different computational tools for route network design and can be classified into two main categories: analytic and network approaches. These approaches differ in their purpose and have different advantages and disadvantages. They should be viewed as complementary rather than alternative approaches (Kuah, 1986).

Analytic models are developed to derive optimal relationships among various components of a transit network system. The analytic approach starts by formulating the design objective as a continuous function of a set of design variables. Assuming that the design variables are continuous, the optimal values of the design variables are then found by using the optimality conditions on the objective function with respect to the individual design variables. The typical design variables are bus route spacing, rail station spacing and service frequency.

The analytic approach is context-dependent in that it requires the shape of the street geometry to be prespecified. The street geometry is usually in the form of simplified networks or regular shaped networks for convenience in formulating models. Some of the geometric network configurations considered are: radial (Byrne, 1975), polar (Vaughan, 1986) and rectangular grid networks (Chien and Spasovic, 2002). Chien and Yang (2000) also considered routes with intersection delays which makes the model more complex but realistic. An example of the street configuration of an analytic model is shown in Figure 2.1. Furthermore, it requires a known demand function which represents the spatial distribution of demand in the service area.

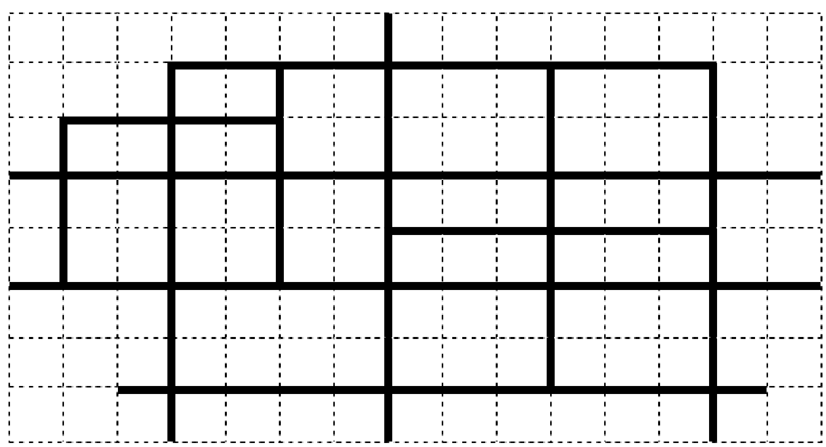

Figure 2.1: Example of an actual street network in an analytic model

While the analytic approach provides approximate solutions using relatively restrictive assumptions regarding the structure of the network, it has the advantage of providing useful relations between design variables and the parameters of the system. These relations can provide useful insights into the interdependencies among the various elements of the problem of bus/rail transit integration. However, this approach is theoretically rigorous and can handle only small sized or regularly shaped networks, since the number of feasible solutions increases drastically with the number of streets. If the street network is oversimplified, the usage of the model developed will be only limited to theoretical studies and may not be applicable to the real world situations.

In the network approach, the service area is represented by nodes and links. The demand is assumed to be concentrated at the nodes and the links represent the segments of the transit routes and usually a travel time or distance is associated with each link. A route is represented by a sequence of nodes. An example of a simple network model is shown in Figure 2.2. An origin-destination (O-D) demand matrix is available to represent the demand by passengers to travel between all pairs of nodes in the network in terms of number of trips made during the selected period of study.

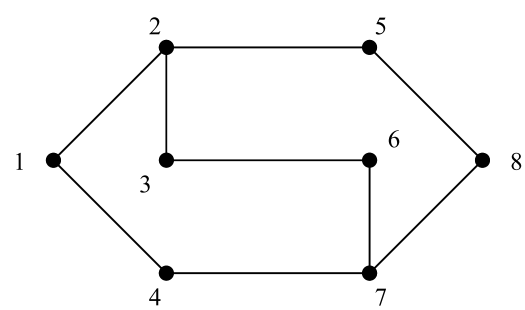

Figure 2.2: Example of a simple network model with 8 nodes and 9 links

Network models are discrete and context free. They are context free in that they do not require one to prespecify the geometry of the street network. Consequently, their applicability is not limited to a particular network structure. They are discrete in that the demand and design elements are represented as discrete variables. It has the advantage of the capability to deal with larger problem sizes and more realistic situations.

The computational tools used to solve both the analytic and network models can be classified into the following types:

1. Conventional mathematical programming methods.
2. Heuristic methods that search for approximately good solutions quickly by using different sets of rules to construct the route network in a step-by-step and iterative procedure (Kuah and Perl, 1989).
3. Hybrid methods that combine the ability of different computational tools to handle complex problems. For example, heuristics can be used for the generation of potential routes and genetic algorithms can be used for the selection of the optimal route set (Pattnaik et al., 1998). Baaj and Mahmassani (1991) also presented a hybrid artificial intelligence/operations research (AI/OR) framework that combines the AI search concepts with OR vehicle routing heuristics.
4. Computer simulation and interactive approach (Rapp, 1972 and Rapp et al., 1976).
5. Metaheuristics such as SA, TS and GA that derive their concepts from mathematical and physical sciences, biological evolution and artificial intelligence (Martins and Pato, 1998; Chien et al., 2001, and etc.).

A considerable amount of effort has been devoted to developing both analytical and numerical techniques for optimization in route network design and they are summarized in Section 2.2.

2.2 Previous Works on Bus Network Design Problems

2.2.1 Analytic approach

Some models developed for bus network design focused only on a limited number of design elements. They were very limited in application because other important design elements such as the route structures were not considered in these models. However, they contributed to the development of the network design problems by providing useful guidelines.

Bus stop models

The bus stop models determined the optimal bus stop spacing of a bus network serving a given level of demand. Lesley (1976) determined the optimal bus stop spacing on a simple, linear bus route so as to minimize the total travel cost. The study assumed that the service area for each bus stop is defined by a circle around each stop with a radius equal to half of the stop spacing. Vaughan and Cousins (1977) determined the bus stop spacing along a linear bus route that minimizes the average travel time for many-to-many demand pattern. By assuming that the daily demand functions vary slowly within stop spacing, Wirasinghe and Ghoneim (1981) determined the optimal spacing of bus stops along a bus route with a non-uniform many-to-many demand distribution by minimizing the sum of operator and user costs. Non-uniformly distributed demand was modeled with cumulative functions of originating and destination trip functions along a bus route. Stochastic stopping of buses was also analyzed.

Bus size models

The bus size models determined the optimal bus size on a bus route serving a given demand. Jansson (1980) optimized bus size and frequency on a simple bus route such that the total cost of operator and user costs was minimized. The optimal bus size was found to increase with the increase in peak passenger flow and in average passenger trip length, and to decrease with the decrease in the average bus occupancy. Oldfield and Bly (1988) determined the optimal bus size for an urban bus service which maximized social benefit. An analytic model was solved to provide an explicit relationship between optimal bus size and factors such as operating cost, level of demand, and demand elasticities. Financial constraints and the effect of fleet size on congestion were also considered.

Zonal transit models

The zonal transit models determined the number of service zones and their boundaries and sizes of each zone in a transit system serving a given demand. Clarens and Hurdle (1975) studied the problem of transporting commuters from a central terminal to various destination zones. Their model determined the optimal zone size and the headway such that the total operating and user costs were minimized. The bus travel time within the service zone was assumed to be directly proportional to the zone area. For the case of fixed demand, the optimal zone size was found to be inversely proportional to the square root of the density of passenger destinations, if the vehicles were filled to capacity, and to the cube root of the density of passenger destinations, if they were not completely filled, under the assumption that problem parameters did not vary over daytime.

Tsao and Schonfeld (1983, 1984) analyzed zonal transit service for a prespecified street network, where the shape of the zone and bus route within zones were clearly defined. The demand pattern was assumed to be many-to-one and demand was inelastic. Tsao and Schonfeld (1983) determined the optimal number of zones (zonal length) and zonal headway of a linear transit route such that the total operator and user costs were minimized. The optimal number of zones was found to be dependent on passenger demand, the value of time and the vehicle operating costs. The optimal headway was found to be dependent on the length of zone in addition to the other three factors mentioned. Tsao and Schonfeld (1984) extended their previous study to the case of two-dimensional, rectangular zones.

Chang and Schonfeld (1993) analyzed bus service zones using the decision variables such as zone size, route length, route spacing and service headway in which supply and demand characteristics were assumed to vary over time. Chang and Schonfeld found that the optimal route length decreased with demand density at a decreasing rate and that the cost functions were U-shaped with respect to the zone elongation parameter, which was defined as the ratio between the length of an area and the width of that area.

Route structure and frequency models

Most analytic optimization models, the route structure and frequency models, considered two decision variables of the route spacing or the location of transit routes and route frequency of a system serving a given demand for optimization. Byrne and Vuchic (1972) determined the optimal bus route spacings and headways on a system of parallel bus routes serving a many-to-one demand to a linearly extended CBD that was perpendicular to the bus routes. The objective of the study was to minimize the total operator and user costs. The model assumed an inelastic demand and that passengers take the least cost route to the CBD and bus operating speeds on each route were equal. An iterative algorithm was developed and the results showed that bus routes should be located such that the total demand from each side of the route was equal. They also found that the optimal headway occurred when the waiting time cost was equal to the operating cost.

Byrne (1975) extended the work of Byrne and Vuchic (1972) to the case of a radial network in which bus routes were radial and were terminating at a common centre, such as the CBD. Byrne showed that the optimal bus routes should be located along the centre line of each sector and should have equal headways. He also showed that the optimum number of lines depended upon the ratio of the access costs to the sum of the waiting and operating costs, if the population density varied only radially. Byrne (1976) extended the work of Byrne (1975) by allowing different bus route operating speeds, which substantially increased the complexity of the model.

Hurdle (1973) also extended the works of Byrne and Vuchic (1972) to consider the design of feeder buses serving a rail line. The rail and bus system were not optimized jointly; instead, only the bus network was optimized. The bus route spacings and bus frequencies were optimized. The objective function was to minimize the total cost function consisting of operator and user costs, but it did not include riding time cost. The demand pattern was assumed to be many-to-one and demand was inelastic. Hurdle showed that the optimal route density and frequency were affected by demand density, bus operating cost and the value of access time.

Holroyd (1967) explicitly considered many-to-many demand patterns for realistic geographic and demand variations. Holroyd analyzed a rectangular grid bus network with a given origin-destination demand matrix that was uniformly distributed over an infinite plane and assuming inelastic demand. The optimal route spacing and headway were determined by minimizing the total cost function consisting of operator and user costs. By analyzing parallel bus routes on a grid street network, Newell (1979) solved the convexity problems and difficulties in determining the optimal bus route location and headway for an idealized service area, while considering many-to-many demand characteristics. Newell showed that the optimal routing would be highly sensitive to the detailed properties of the trip distribution and the cost parameters. Newell also found that a square grid of straight line bus routes was not likely to be an optimal geometry even under highly idealized conditions.

Instead of assuming inelastic demand as in the previous models, Kocur and Hendrickson (1982) considered the design of a local bus transit network with demand elasticity such that ridership was sensitive to bus service quality and fares. The travel demand pattern was many-to-one and their model optimized bus route spacing, bus headway and fare for a bus system operated on a rectangular grid. Kocur and Hendrickson considered three objectives: profit maximization, net user benefit maximization, and combined operator profit maximization and net user benefit maximization. For all the objectives, the study found that optimal route spacing and headway are proportional to one another.

Vaughan (1986) analyzed a polar bus network consisting of ring and radial routes to determine the optimal spacing and headway by minimizing user travel time subject to a fleet size constraint. A fixed demand and a many-to-many travel demand was described by a continuous function of the positions of a commuter's home and workplace. It was also assumed that buses travel at a constant speed. The results indicated that under optimal conditions both spacing between routes and headway between buses should be inversely proportional to the cube root of the proportion of commuters joining and leaving the route. Chang (1991) developed a model to optimize radial bus networks, in which multi-periods and many-to-many demand patterns were considered. The total cost function consisting of operator and user costs was minimized and inelastic demand was assumed.

Chang and Schonfeld (1991) developed analytic models for parallel route bus systems with time dependent, many-to-many and elastic demand which varied in different time periods. With some approximations, closed form solutions for the optimal route spacings, headways, for different time periods and stop spacings were obtained for inelastic demand conditions. Later, Chang and Schonfeld (1993) modeled parallel route bus systems with time dependent, many-to-many and elastic demand distributions, in which the vehicle operating cost varies in different time periods and along one spatial dimension. Spasovic and Schonfeld (1993) developed a model to optimize service coverage for a rectangular urban area using uniform and linearly decreasing density functions, and assuming inelastic demand. The optimal route length, headway, route and stop spacing were found by minimizing the total cost function consisting of operator and user costs. Using a sequential substitution algorithm, Spasovic, Boile, and Bladikas (1994) optimized bus transit service coverage by maximizing operator's profit and social welfare. Optimal route spacing, operating headway and fare were determined, while considering the impact of realistically irregular geographic, socio-economic, demand and traffic characteristics. Chien and Schonfeld (1997) optimized a grid transit system in a heterogeneous urban area considering two-dimensional realistic geographic conditions without oversimplifying the spatial and demand characteristics. The route spacings and headways were optimized by minimizing the total system cost function consisting of operator and user costs with fixed demand and many-to-many travel patterns.

Imam (1998) demonstrated optimal design of a bus transit system analytically for the case of variable demand with transit ridership sensitive to service levels. It was assumed that transit passengers always used the nearest available transit route and had the choice of using the transit mode or other alternative modes. Their transport mode choice was dependent on the five variables: wait time, walk time, in-vehicle travel time, fare, and auto time and cost, represented by their respective demand mode choice coefficients. The major decision variables considered here were the spacing between parallel bus routes, the frequency of service on a route and the fare, with an objective function of profit maximization with or without vehicle capacity constraints. Optimal spacing between parallel bus routes and headway were found to be proportional to each other for the same objective function with or without vehicle capacity constraints. The optimal values with or without capacity were based strongly on the demand mode choice coefficients of walk and wait time. Similar relations had also been obtained by Hurdle (1973) and Kocur and Hendrickson (1982) for different types of transit systems. On the other hand, optimal fare level without vehicle capacity constraint was extremely sensitive to changes in the square of the mode choice coefficient of the wait and walk time. Similar relations had been obtained by Kocur and Hendrickson (1982) in the case of optimal spacing between parallel bus routes and headway. In vehicle capacity constraints, the mode choice coefficient of the wait and walk time did not affect the fare level.

Chien and Spasovic (2002) proposed an analytic model which was able to optimize the design of urban bus transit systems in a realistic geographic environment with irregular variations in land use, travel speeds, operating costs and other such variables. Elastic demand sensitive to the service quality variables, such as travel time (e.g. access, wait and in-vehicle times) and fare and many-to-many demand patterns were also considered. This could be done without oversimplifying the spatial characteristics and demand patterns by subdividing the irregular service region into arbitrarily small zones based on the different internal characteristics, such as the demand densities and travel speeds, reflecting the fact that each zone could have different land use and traffic characteristics. The objective functions included maximization of total operator profit and social welfare functions. The total profit was defined as the total revenue minus the operator cost, while the total social welfare was consumer surplus plus the operator's profit. The main decision variables to be optimized were the route spacings, operating headways and fare. Results showed that both the objective total profit and social welfare functions were relatively flat near the optimum with respect to route spacing and headway. This indicated that minor changes in the solution away from the optimum would not change the profit significantly and allowed bus operators considerable flexibility in fitting the route structure to local circumstances.

2.2.2 Network approach

The literature on network approaches for bus network design can be classified into three groups: headway models, route structure models, and combined headway and route structure models. While the purpose of the headway models is to determine the optimal bus route frequencies and the route structure models is to determine the location of routes in a bus network, these two models are limited as they consider only one aspect of the bus network design problem. Moreover, they assume independence between route structures and headways. However, their simplicity allows the optimization model to be solved exactly or using a heuristic algorithm to provide good near-optimal solutions.

Headway models

Lingaraj et al. (1976) developed a model for determining the headways that minimized the system operating cost on a prespecified route network using the chance constrained programming techniques. Erlander and Scheele (1974) determined optimal bus frequencies and passenger flows, where the objective function was to only minimize the travel time of passengers. Scheele and Erlander (1980) developed an iterative algorithm and tested it on an actual bus network in the town of Linkoping in Sweden. The results produced by the model suggested certain actions which were in agreement with the actions taken by the bus operator in practice.

Route structure models

Rapp (1972) and Rapp et al. (1976) developed the interactive graphics of transit design system (IGTDS) and interactive graphics network optimization system (TNOP) for determining the route structure of an urban transit network serving a many-to-many demand pattern. The iterative procedure allowed planners to scan the solution space "thoroughly" and be able to determine a good solution. Hsu and Surti (1975, 1977) developed bus routes in a given service area using a heuristic approach of insertion of nodes. The procedure started by identifying potential nodes in the service area and then adding routes successively to connect the nodes into an emerging transit network. An evaluation system with operating revenues, accessibility and connectivity as criteria was developed to help planners in evaluating the connected network. Hsu and Surti (1977) also developed the decomposition approach for Denver's urban area. Mandl (1980) developed a heuristic procedure to improve the route system of an existing transit route network. The procedure started with a feasible set of routes (i.e. the existing route structure) and then performed a search to find a route structure that would reduce the average passenger's transportation cost. Bansal (1981) formulated an optimization model to determine a set of bus routes for serving a many-to-many demand pattern. The objective function minimized the total operating cost and travel cost, and the model was solved heuristically. The study showed that for any network, a close-to-optimal solution was one with the minimum number of routes joining all O-D pairs by shortest paths.

Combined headway and route structure models

Early combined route structure and frequency models are two-stage heuristic models in which transit routes are built in the first stage and service frequencies are determined in the second stage. Lampkin and Saalmans (1967) developed a heuristic model to design a complete transit route network by inserting nodes to a base network such that all the demand was satisfied. The model then assigned an operating frequency to each route of the complete network so as to minimize total travel time.

The technique assumed a fixed passenger demand and did not consider any restriction on bus fleet size. Silman et al. (1974) developed a two-phase method similar to that of Lampkin and Saalmans (1967). In the first phase, bus routes were generated by a process of insertion and deletion based on the criteria of total travel time. In the second stage, the optimal route frequencies were determined such that the sum of the journey time plus discomfort penalties was minimized. Rea (1972) also developed a heuristic approach for the design of routes and frequencies using service specification models in which links with low flow that did not warrant transit services were eliminated from the initial network.

The combined models described above implicitly assume that the route structure and frequency are independent. This assumption is inaccurate as we expect the two design elements to be interdependent. The models below are all able to optimize the route structure and frequency simultaneously.

Baaj and Mahmassani (1991) discussed the sources of complexity of the transit network design problem and existing solution approaches to this problem and their limitations. They developed an AI-based approach, which used expert knowledge to guide AI search techniques and produce superior solutions efficiently. They also presented the framework of an AI/OR hybrid solution approach that combines AI search concepts with vehicle routing heuristics and transit system analysis methods. The framework consisted of three major components: (1) The Route Generation Design Algorithm (RGA) which generated different sets of routes corresponding to different trade-offs among the principal objectives; (2) The Analysis procedure (TRUST) which was described in detail in Baaj and Mahmassani (1990). TRUST assigned the demand matrix to the transit network and computed the corresponding user and operator costs as well as the pertinent measures of system performance; and (3) The Route Improvement Algorithm (RIA) which considered the set of routes generated and utilized the results of the analysis procedure to generate an improved set of routes. The hybrid approach was able to incorporate the knowledge and expertise of transit network planners together with efficient search techniques.

Baaj and Mahmassani (1995) focused on the RGA for the design of transit route networks. This algorithm was one of three major components of an AI-based hybrid solution approach to solving the transit network design problem (Baaj and Mahmassani, 1991). RGA was a design algorithm heavily guided by the demand matrix and allowed the designer's knowledge to be implemented so as to reduce the search space. It was also able to generate different sets of routes corresponding to different trade-offs among conflicting objectives (user and operator costs). Numerical experiments were conducted to test the performance of RGA on data generated for the transit network of the city of Austin, Texas.

Pattnaik et al. (1998) studied urban bus route network design using a genetic algorithm as the tool for selecting the optimal route network to minimize the overall operator and user costs. The GA model for the route network design was carried out in two phases. The first phase was the generation of a large set of potential routes called the candidate route set by a candidate route generation algorithm (CRGA) that took care of the network specific characteristics and the travel demand. The second phase involved the selection of the optimum solution route set with their frequencies from the candidate route set using the GA. Two methods based on the binary coding scheme were implemented: the fixed string length coded (FSLC) model and the variable string length coded (VSLC) model. The FSLC model worked with a fixed size of the number of routes in a feasible solution. Since the number of routes in the optimum solution was unknown, the evaluation was carried out by assuming a route set size in each iteration and tested a range of route set size values, in order to find the optimal route set size. It was simple to code this model but it required many iterations and excessive computational effort to generate the solution. For the VSLC model, the variation for the solution route set size was incorporated in the coding itself and was optimized by GA. The route set and the number of routes were optimized simultaneously, thus avoiding any extra iteration for the number of routes. However, it required more complex coding than the FSLC. When applied to the case study network of Madras Metropolitan City, South India, a comparison of the performance of the two models showed that FSLC gave a better solution in terms of lower objective function value, unsatisfied demand, total travel time, bus kilometre and fleet size, while VSLC resulted in a smaller number of routes and less waiting time. FSLC required more computation time than VSLC mainly due to the iterations being carried out for the route size. Hence, the VSLC model can be adopted as a design procedure which gives a near-optimal solution with considerably less computation time. FSLC would be slightly superior to VSLC if computation time was not a constraint.

Tom and Mohan (2003) adopted the objective function of minimizing the total system costs comprising of the sum of the operating cost and the user's travel cost (Baaj and Mahmassani, 1995). They found that the previous two coding schemes proposed by Pattnaik et al. (1998) required an assumption on initial frequency and then allocating the demand, followed by frequency computation iteratively to arrive at the final frequency of the selected routes. To avoid the iterations on the frequency, a third coding scheme, the simultaneous route and frequency coded (SRFC) model was proposed that incorporated the frequency of the route as a variable in addition to the route details. A sample study on the Chennai Metropolitan City, South India, network showed that when compared with the FSLC and VSLC models (Pattnaik et al., 1998), SRFC gave the best solution in terms of lowest objective function value, least operating cost, highest direct demand allocation, lowest fleet size, and was considered superior to the other two models.

Ngamchai and Lovell (2003) designed the bus route configuration and the service frequencies on each bus route with the objective of minimizing the total operator and user costs. They mainly focused on determining the transfer locations in the network and coordination of headways by the ranking of transfer demands at the transfer terminals in a three-step process: (1) The route generation algorithm to construct an initial feasible set of routes, (2) The route evaluation algorithm to design the service frequency on each route and apply headway coordination techniques at transfer points to decrease transfer times between routes, and (3) The route improvement algorithm to improve the efficiency of the network using genetic operators and changing the transfer locations in the system. In addition, they suggested that when the chromosome was binary-coded (Pattnaik et al., 1998), the search was merely a random choice of a new chromosome and undirected. In their proposed model, integer representations were coded in GA to represent the set of routes in a solution and it was proven to be more efficient than the binary-coded GA. Seven new problem-specific genetic operators were also proposed to handle the problem and the relative efficacy of each genetic operator were analyzed.

Chakroborty and Dwivedi (2002) also proposed a three-step iterative process based on the principles of genetic algorithms to determine an "optimal" or "efficient" transit route network for a given road network and transit demand data with respect to certain criteria. Firstly, various initial reasonable route sets for the given road network and demand matrix were determined using the Initial Route Set Generation (IRSG) procedure. The evaluation step and the route modification step were executed repeatedly until an "optimal" or "efficient" route set was obtained. Instead of evaluating the quality of the solution based on just the objective function value, the evaluation scheme consisted of a set of criteria which gave a measure of the goodness of a route set. Lastly, the route sets were modified using genetic algorithm to evolve into a better group of route sets. The performance of the proposed methodology was tested using a real-world benchmark Swiss network initially used by Mandl (1980) and later by Baaj and Mahmassani (1991) to test their route design algorithms. Mandl (1980) also provided the link travel time data and the complete transit demand matrix for the network. The results showed that the proposed method performed substantially better than the existing procedures.

2.3 Previous Works on Feeder Bus Network Design Problems

2.3.1 Analytic approach

Wirasinghe (1977) and Wirasinghe et al. (1977) designed a coordinated rail/bus transit system that served peak travel between a metropolitan region and its CBD. The rail lines were assumed to be radial. Wirasinghe (1977) optimized the zone boundary from which feeder-buses should serve a rail line so as to minimize the total user and operator costs for a given set of rail station spacings and constant train headways. Wirasinghe et al. (1977) extended this work to optimize additionally the rail station spacings and the train headways. Wirasinghe and Ho (1982) designed an optimal commuter bus system serving peak demand for “drive and ride” travel between the CBD and residential areas. Passengers were assumed to access the radial bus routes by auto and then travel to the CBD by the commuter bus system. The study determined the optimal bus route density and headway to minimize the sum of operator and user costs. The optimal route density was found to be insensitive to the peaking effect of demand over time and to small changes in the route frequency. Wirasinghe (1980) analyzed the case of feeder bus access to a rail line on a rectangular street network. Feeder buses along parallel routes fed passengers to a linear rail line that ran perpendicular to the feeder routes toward the CBD. The study determined the bus route density, bus frequency and rail station density that minimized the total operator and user costs. A major assumption was that all passengers walked to the nearest bus route. Hurdle and Wirasinghe (1980) extended the study of Wirasinghe (1980) to include several feeder modes such as auto, bus and bicycle. However, only the rail station spacing was optimized.

Kuah and Perl (1988) found that the above studies of feeder bus network design had several limitations. First, they dealt with the problems of determining the optimal stop spacing separately from that of determining optimal route spacing and operating headway. They had also considered only the case of many-to-one demand pattern. Kuah and Perl presented an analytic model for the design of an optimal feeder bus network for accessing an existing rail line, which avoided the sequential approach and optimized the stop spacing, route spacing and headway simultaneously. The study assumed inelastic demand and many-to-many travel pattern, and minimized a total cost function consisting of operator and user costs. The results with regards to bus route spacing and headway were similar to those obtained previously in the sense that the route spacing and operating headway were not highly sensitive to changes in the relevant system parameters. Numerical examples indicated that the proposed model had provided reasonable solutions.

In addition, the problems in existing studies are solved analytically by simplifying the route structure or demand distribution of the service area. Chien and Schonfeld (1998) presented an optimization model for jointly optimizing the design variables of an integrated rail transit system and its associated feeder bus system in a rectangular service area with irregular and discrete realistic geographic and demand variations along one dimension, i.e. along the rail corridor. The travel demand pattern was many-to-many and the decision variables included rail line length, rail and bus headways, station and stop spacings, and bus route spacing. The objective was to minimize the total cost including the supplier and user cost. Besides deriving the analytic relations among variables, they also developed a successive substitution algorithm for optimizing the key variables of transit systems. This algorithm was able to converge quickly towards the minimum of the objective function. It was found that the total cost function is relatively flat near the optimum and minor changes in the solution from the optimum would not change the cost significantly.

Chien and Yang (2000) optimized the location of a feeder bus route and its operating headway to provide service for connecting an irregular shaped suburban area with a generally irregular grid street pattern and a CBD or a major intermodal transfer station.

Subject to geographic, capacity and budget constraints, the objective function of total cost including supplier and user costs had been minimized, while considering intersection delays, heterogeneous demand distributions and realistic geographic variations. The travel demand pattern of the area was many-to-one and the demand was assumed inelastic with respect to service quality or price. The exhaustive search algorithm (ES) proposed could find the optimal feeder bus route location and headway on a given network because the optimal solution was identified by evaluating all feasible solutions found. However, if the given network was more complicated in terms of greater number of the streets, the number of feasible routes would become huge and result in a drastic increase in computation time. Although the near-optimal algorithm could improve the computation efficiency, it sacrificed the accuracy of the results.

Since the ES algorithm was computationally expensive to solve the route design problem, Chien et al. (2001) extended Chien and Yang's work (2000) by presenting a genetic algorithm to solve the problem. The GA started with an initial population randomly generated by the route generator followed by route improvement by GA. The results showed that the optimal solutions found by ES and GA were identical. However, the computational time for GA was significantly less than that for ES, especially for large and complicated networks. Thus, Chien et al. had demonstrated that the proposed GA efficiently converged to the optimal solution, validated by applying ES.

In an intermodal transit system in which the rail line and its feeder bus routes are integrated, passengers may need one or more transfers to complete their journey.

Long transfer times may significantly deteriorate the service quality. Therefore, effective intermodal coordination can reduce transfer times and contribute significantly to service quality improvements. Such problems are known as transfer optimization problems. Chowdhury and Chien (2002) developed a method to seek better coordination for an intermodal transit system using an analytical approach. The objective was to determine the optimal headways for all transit routes and slack times for all coordinated routes by minimizing total cost, including supplier and user costs. A four-stage procedure was developed and a numerical search algorithm (Powell's algorithm) was applied to solve this problem.

2.3.2 Network approach

Kuah and Perl (1989) was the first to attempt to define the FBNDP using a mathematical programming model based on the network approach. The FBNDP was to design a feeder bus network to access an existing rail system. It was considered under two different demand patterns, many-to-one (M-to-1) and many-to-many (M-to-M). He showed that the mathematical programming model for the M-to-1 FBNDP could be generalized to the M-to-M FBNDP. The model optimized the route structure and the operating frequency with the objective function of minimizing the sum of user and operator costs. A heuristic model was presented, which generalized the 'savings approach' to incorporate operating frequency. The computational analysis showed that the proposed heuristic provided reasonable feeder bus network solutions that are superior to manually designed networks.

Martins and Pato (1998) extended the works of Kuah and Perl (1989) by designing better heuristics to improve previously obtained solutions. They built the initial solution using the sequential savings proposed by Kuah and Perl (1989) and also developed a two-phase method to generate the initial solution. The initial solution was improved using some heuristic procedures, as well as tabu search heuristics with different strategies. In addition, they also generated a set of problems simulating real life situations for a more detailed computational study. The results showed that even the simplest short-term version of tabu search produced very good solutions and they concluded that it would be one of the most promising heuristics in the future.

Shrivastav and Dhingra (2001) proposed the development of a heuristic algorithm for feeder routes of public buses to suburban railway stations. Mumbai, India was used as a study area. This heuristic algorithm was the first part of a model that had been developed for operational integration of suburban railway stations and public buses. The second part was the determination of optimally coordinated schedules of feeder bus services for the existing schedules of suburban trains. The proposed heuristic feeder route generation algorithm (HFRGA) was heavily guided by the demand matrix similar to Baaj and Mahmassani (1995). It adopted different node selection and insertion strategies by giving priority to nodes having higher demands over low demand nodes and therefore, insertion of high demand nodes were done first. The proposed algorithm was able to develop feeder routes that could satisfy demands to various nodes without any transfer with reasonable acceptable delays to high demand nodes. It was also able to identify the best route for each node among all available routes and have it inserted in the best possible way in the selected route. The results showed that the developed feeder routes were much fewer in number as compared to the existing ones and would reduce the fleet size for the same demand satisfaction.

2.4 Detailed Description of Previous Approaches to FBNDP

Due to the complex and combinatorial nature of the FBNDP, the past approaches have used heuristic algorithms to solve the problem. A heuristic algorithm can search for a near optimal solution rather than the best solution at relatively low computational time. The existing heuristics proposed for the FBNDP solved the problem by first building an initial feasible solution starting from scratch using a construction algorithm and next to improve the current solution using a local search algorithm.

Construction algorithms start from an empty solution and iteratively add components to the partial solution, often in a greedy manner, until a complete solution is obtained. An example of a greedy construction heuristic for the travelling salesman problem (TSP) is the *nearest neighbour* construction heuristic. Starting from an initial city, this algorithm builds a tour by always choosing to go to the nearest unvisited city before returning to the initial city. Construction algorithms are fast to build and always give a fairly good approximation compared to randomly generated solutions.

The results produced by construction algorithms are often far from optimal and they can be further improved by local search algorithms. Local search algorithms start from a complete initial solution and try to search its neighbourhood for a better solution. If such a solution is found, it becomes the current solution and the local search continues. When no more improving neighbour solution is found, the algorithm ends in a local optimum.

Since there are relatively few papers published on FBNDP and because of the similarity between the FBNDP and the multi-depot vehicle routing problem (MDVRP), which is the problem of designing a set of delivery routes from several depots to a large number of demand points so as to minimize the total route distance, the methods for developing heuristics for the FBNDP are adapted from the existing heuristics for the MDVRP. However, the FBNDP has the added complexity of a non-linear objective function to minimize both operator and user costs, while the MDVRP only has to minimize operator cost (travel distance). In addition, the operating frequency in the FBNDP is a decision variable while in the MDVRP, it is predetermined.

Some of the construction heuristics for building initial solutions are the sequential building and two-phase building methods. The local search improvement methods include the displacement and exchange heuristic and tabu search. The details of their implementation are described in Section 2.4.1.

2.4.1 Route construction heuristics

Sequential building

The sequential building heuristic is proposed by Kuah and Perl (1989), which is adapted from the 'sequential savings approach' for the MDVRP.

The algorithm starts by calculating, for each stop *i*, the cheapest direct route cost of linking stop *i* to station *j*, which is given by

$$TC_i^j = C_{js}q_i + 2\sqrt{\lambda_0\lambda_w l_{ij} q_i} + \frac{\lambda_r}{U} l_{ij} q_i \qquad (2.1)$$

The construction procedure initiates the first route by selecting the stop with the highest direct route cost, and linking it to the nearest station. A route is then constructed by inserting subsequent unassigned stops into the current route, between the stop that is first placed and the station that it is linked to, based on the *maximum savings* (MSVTC) value.

The savings from including an unassigned bus node *i* between points *p* and *q* of an emerging route is given by

$$SAVTC_i(p,q) = TC_i^j - INCOST_i(p,q) \qquad (2.2)$$

where the insertion cost $INCOST_i(p,q)$ is the cost of inserting stop *i* between stops *p* and *q*. The insertion cost is the difference between the total route cost before and after the insertion, and is given by

$$INCOST_i(p,q) = C_{js}q_i + 2\sqrt{\lambda_0\lambda_w}\left(\sqrt{L_k^a (Q^k)^a} - \sqrt{L_k Q^k}\right) + \frac{\lambda_r}{2U}\left(L_k^a (Q^k)^a - L_k Q^k\right) \qquad (2.3)$$

where the superscript ‘a’ denotes the value after insertion.

However, as discovered by Tillman and Cain (1972), this savings equation is only applicable to the case of a single station. They introduced the concept of ‘modified distance’ for the case of multiple stations. Because of the multi-station nature of the

FBNDP, a 'modified total cost' (MTC) of a direct route from stop i to station j is obtained as follows

$$MTC_i^j = \underset{j' \in R}{\min imum}\ TC_i^{j'} - \left(TC_i^j - \underset{j' \in R}{\min imum}\ TC_i^{j'} \right)$$

$$= 2\ \underset{j' \in R}{\min imum}\ TC_i^{j'} - TC_i^j \qquad (2.4)$$

Thus, the *modified savings* measure is

$$MSVTC_i(p,q) = MTC_i^j - INCOST_i(p,q)$$

$$= \left(2\ \underset{j' \in R}{\min imum}\ TC_i^{j'} - TC_i^j \right) - INCOST_i(p,q) \qquad (2.5)$$

The building of a route ends when at least one of the following conditions occurs:

- When the current emerging route cannot be expanded because the feasibility constraints are violated.
- When the 'savings' from expanding any of that route is smaller than a prespecified value.

If the building has ended due to the first case, a new route is initiated by selecting the unassigned stop with the greatest direct route cost. If the building has ended due to the second case, a new route is initiated by selecting the bus node with minimum MSVTC value.

Two-phase building

The two-phase building heuristic is suggested by Martins and Pato (1998) which splits the stops into *border* and *non-border* stops. The criterion for sorting the stops is by calculating the ratio between the direct route cost of the cheapest station $j*$ and the same cost for the next cheapest station $j**$ for each stop:

$$r_i = TC_i^{j*} / TC_i^{j**} \tag{2.6}$$

A classifying parameter η = 0.85 was chosen. If $r_i \leq \eta$, linking stop i directly to station $j*$ is far more cheaper than linking it directly to station $j**$. Then, linking stop i to station $j*$ is considered as a more attractive choice. Stop i is classified as a *non-border* stop and is assigned to station $j*$. If $r_i > \eta$, then stop i is classified as a *border* stop and is not assigned to any station because linking to either stations will not cause much difference in the total cost.

After the stops have been classified, the route building procedure works in two phases. In the first phase, border stops, which are not assigned to any stations, are linked to stations by a route-building procedure identical to the one applied in the sequential building heuristic. At the end of this phase, all border stops would have been included in the routes constructed.

In a second phase, the remaining non-border stops, which have already been assigned to stations, are inserted to the existing routes ending at the station to which they were previously assigned. If some stops have still not been included, new routes are then created.

By separating the stops into two subgroups, the two-phase building heuristic requires significantly lesser computations then the sequential building heuristic. In the paper stated, the results for the initial solution do not differ much from the sequential building heuristic.

2.4.2 Local search heuristics

Displacement heuristic

In the algorithm proposed by Kuah and Perl (1989), after solving the TSP using the 2-opt procedure for each route, the initial solution is improved using the *displacement procedure*. The displacement procedure considers two types of displacements: an *internal* displacement, where a stop is displaced to another route linked to the *same* station, and a *non-internal* displacement, where a stop is displaced to a route linked to a *different* station.

Firstly, only internal displacements are performed. This is done by constructing a *displacement list* for every station, which records and ranks all the internal feasible displacements that would result in a reduction of the total cost. The displacement that gives the largest reduction is performed and the displacement list for that station is updated. The process continues until the displacement lists of all stations are empty. Next, a displacement list which records and ranks all non-internal feasible displacements is constructed. Similarly, a displacement that gives the largest cost reduction is performed and the list is updated. The procedure terminates when the displacement list is empty.

Exchange heuristic

In another paper by Martins and Pato (1998), the exchange of positions of two stops from two *different* routes is also performed if there is a reduction in the total cost, while ensuring that the new routes remain feasible. As in the displacement heuristic, a ranked list of the exchanges is used and the heuristic terminates when the list is empty.

Tabu Search

Martins and Pato (1998) also attempted to solve the FBNDP using TS. For their basic TS, the move attributes are defined as the stops before and after the stops that are being displaced or exchanged. For example, if stop i is between stops s and t before displacement, then the move attributes are s and t. The tabu restrictions imposed will be to prevent the replacement of a stop in its previous position for some iterations, with the tabu list size being $\sqrt{I+J}$. Hence, when displacement takes place, if stop i is between stops s and t before displacement, then after a displacement takes place, stop i cannot be re-inserted between s and t for a pre-specified number of iterations. If an exchange takes place, an additional restriction will be imposed on the other stop, preventing it from re-inserted between its previous attributes.

Other characteristics of its basic TS include:

- Using the best solution aspiration criterion to override the tabu status;
- The selection criterion for choosing moves is either the “first best admissible” or the best admissible, giving rise to two versions of basic TS;
- A stopping criterion, defined as the number of iterations without improvement of the objective function, is set equal to $\sqrt{I+J}\sqrt{K}$ (where K is the total number of routes).

Their intensification strategy is employed on a short-term basis by forbidding the stops from leaving the routes of the best-so-far solution. It serves to intensify the search in a region of good solutions before leaving the search to take other directions. Therefore, if the best-so-far solution is found as a result of a displacement of stop *i* from route *k1* to the place between stops *p* and *w* in route *k2*, all solutions that do not have stops *p*, *i* and *w* sequentially in a route will be forbidden during $\sqrt[4]{I+J}$ iterations. This also applies to exchange moves.

The diversification strategy is a longer term memory search process which penalizes the less frequently changed attributes and is carried out after δ iterations without improvement in the objective function. The most constant linkages are penalized according to

$$L_{ih}^{'} = L_{ih} + (P_{ih} / TIT)\sqrt[4]{\underline{L}} \qquad (2.7)$$

where L_{ih} is the actual distance between stops *i* and *h*, $L_{ih}^{'}$ is the penalized distance between stops *i* and *h*, $\underline{L}$ is the average distance between any two points, P_{ih} is the number of iterations with the linkage from *i* to *h* present, and TIT is the total number of iterations from the beginning of the previous run of the basic TS.

2.5 Concluding Remarks

In this chapter, we summarize the previous works for the bus and feeder bus network design problems according to their approach for solving the problem: the analytic approach and the network approach. In general, two observations can be deduced from this literature survey. Firstly, it seems that there is no standard definition of the network design problem. As a result, different studies have focused on various aspects of the problem, defining different objective functions and constraints, different decision variables and making different assumptions to simplify the problem, hence making critical comparison a difficult task. However, such diversity can be understandable because public transportation systems in different countries have different sets of rules and objectives to satisfy, and passengers have demand for different types of services depending on the zones where they live in, their lifestyle and daily routines and their commuting behaviour.

Secondly, the literature survey also reveals that that there has not been substantial work for both the analytic and network approaches to the network design problem, especially in the area of multiple-mode transit system in which the feeder bus system provides service for an existing rail network. Most of the existing studies have been focused on the design of a single-mode network. As such, our research is focused on the development of new approaches for the feeder bus network design problem.

Chapter 3 Approach to Solving FBNDP

In this chapter, we present our approach for solving the FBNDP. First, the route construction heuristic for obtaining an initial solution is described, followed by defining the neighbourhood solutions and introducing the metaheuristics used for improving the initial solution. Lastly, the test problems which are used to test the performance of the metaheuristics are described.

3.1 Generating the Initial Solution

The initial solution is constructed on a stochastic basis. Each route is built sequentially as follows: first, a station is selected at random, then, stops, chosen at random, are added to the route linking to this station. Each time a stop is added, the length of the route is checked. When it exceeds the maximum route length, the current route is terminated and a new route will be built in the same way. The procedure continues until all the stops have been included in the routes.

However, random selection of stops without any restrictions may cause too many bad selections and result in a poor initial solution. Thus, the concept of *delimiter* proposed by Van Breedam (2001) is applied here. A delimiter is a restriction in terms of the distance between the stations and the stops to be selected at random. The delimiter is calculated as follows:

For each stop i, determine the distance of its nearest station j:

$$\text{Dist}_i = \min_j d_{ij} \tag{3.1}$$

The *initial delimiter* D_I is equal to the maximum of the set of minimum distances calculated:

$$D_I = \max_i (\text{Dist}_i) \tag{3.2}$$

Thus, the distance between the random stops and the stations selected must be less than or equal to D_I, or else a new stop will be generated. In this way, the delimiter will prevent linking a stop and a station that are too far apart such that they are unlikely to form a good solution.

3.2 The Metaheuristics Approach

Metaheuristics are generalizations of local search algorithms. They are the most recent development for solving hard combinatorial optimization problems by incorporating concepts derived from artificial intelligence, biological evolution, mathematical and physical sciences to improve their performance. In this research, four metaheuristics, namely SA, TS, GA and ACO have been developed and applied to optimize the FBNDP. Their historical developments and general descriptions are described in details in the later chapters. In general, all these four metaheuristics are unique in their own ways when searching for the optimal solution. SA is able to avoid getting trapped at a local minimum by probabilistically accepting a neighbourhood solution that is worse than the current one. TS makes use of short and long term memory structures to store certain attributes of recently visited solutions in tabu lists and forbid these attributes for some time to prevent the repetition of moves. In GA, new and better generations of solutions are evolved by going through the selection, crossover and mutation processes. ACO is inspired by the behaviour of real ants which lay pheromone trails on the paths they have travelled to direct the search by the future ants. Each artificial ant is capable of constructing a feasible solution on its own using a probabilistic decision rule that makes use of pheromone trails and priori information about the problem. These metaheuristics have proven to be very efficient and effective in solving large and complex combinatorial optimization problems, such as the vehicle routing problem (Van Breedam (1995, 2001), Bullnheimer et al. (1999a, 1999b, 1999c), Tan et al. (2001a, 2001b)). Their concepts and applications are also widely described by Reeves (1993), Osman and Kelly (1996), Aarts and Lenstra (1997), Sait and Youssef (1999) and Pham and Karaboga (2000).

3.3 Defining Neighbourhood Moves

For the application of metaheuristics such as SA and TS, the choice of an appropriate neighbourhood structure is crucial for their performance. The neighbourhood structure defines the set of solutions that can be reached in one single move from a current solution.

An example for the TSP is the *k*-opt neighbourhood in which neighbour solutions differ from the current solution by deleting *k* arcs and reconnecting it by *k* others. The 2-opt algorithm tests whether the current tour can be improved by replacing two edges. The 2-opt algorithm is included in the solving of FBNDP and is carried out after the initial solution has been constructed and whenever a new best solution has been found. The purpose is to reorder the stop sequence in every single route to reduce the route distance, which in turn reduce the total cost, before the local search is continued. The use of the 2-opt algorithm is justifiable because the size of the problems tested in this research is small.

In order to obtain a neighbourhood solution for the FBNDP, several move types are defined to describe the way stops are relocated between the routes. It is assumed that only feasible moves with respect to the constraints are performed. Three types of moves are considered here.

Exchange is a move in which the positions of two stops from two different routes are exchanged.

Reduction-displacement is a move of a stop from one route to another existing route. The purpose of this type of move is to reduce the number of routes.

Addition-displacement is the removal of a stop from one route and allowing it to form a new route by linking it to its nearest station. Since the focus of this problem is on minimizing the total cost and not the number of routes (number of vehicles required), a solution with the minimum number of routes may not give the smallest objective value. The intention of this move is to increase the number of routes whenever there is a chance of further reducing the objective function.

3.4 Description of the Test Problems

Our experiments are based on two types of problem data as described below.

3.4.1 The base problem

The benchmark problem, which is named as the base problem, is taken from the literature (Kuah and Perl, 1989) and shown in Table 3.1 and Table 3.2. It includes 55 stops and 4 stations (one of which is specified as the destination station), covering a service area of 2 × 2.5 square miles. The bus-stop density is 11 stops per square mile, with hourly demand density of approximately 2200 passengers per square mile, i.e. a constant demand of 200 passengers per hour at each stop. The values for the parameters such as the bus operating cost, rail user cost, riding-time cost, waiting-time cost, maximum allowable route length, bus capacity, bus operating speed, and maximum available seat-hours are shown in Table 3.3.

Table 3.1: Bus stop locations

Stop index	X-coordinate	Y-coordinate
1	30	234
2	62	235
3	119	250
4	182	249
5	134	228
6	163	230
7	115	222
8	87	215
9	24	203
10	60	193
11	125	197
12	150	210
13	183	196
14	108	186
15	85	177
16	37	169
17	130	173
18	185	164
19	12	163
20	67	153

21	105	157
22	123	152
23	32	133
24	55	135
25	73	135
26	89	144
27	142	137
28	161	143
29	18	107
30	46	107
31	107	115
32	147	117
33	172	124
34	31	95
35	91	103
36	113	99
37	13	80
38	66	87
39	83	83
40	141	92
41	167	97
42	67	65
43	122	75
44	150	67
45	177	68
46	95	59
47	17	47
48	47	43
49	130	48
50	71	35
51	108	33
52	169	35
53	13	25
54	35	17
55	63	7

* Distances are specified in hundreds of miles.

Table 3.2: Station locations

Station index	X-coordinate	Y-coordinate
56 (Destination)	42	72
57	78	116
58	123	137
59	160	178

* Distances are specified in hundreds of miles.

Table 3.3: Values for parameters

Descriptions	Units	Value
Bus operating-cost, λ_0	\$/vehicle-mile	3.0
Rail user-cost, C_{js}/mile	\$/passenger-mile	0.15
Riding-time cost, λ_r	\$/passenger-hour	4.0
Waiting-time cost, λ_w	\$/passenger-hour	8.0
Max. allowable route length, D_k	Mile	2.5
Bus capacity, c	Seat	50
Bus operating-speed, U	Mile/hour	20
Max. available seat-hours, ASH	Seat-hours	5500

3.4.2 The randomly generated test problems

The second set of test problems consists of 20 randomly generated test problems to further compare the performance of the metaheuristics proposed. These problems vary in several characteristics, such as the problem size and the problem structure.

The problem size is determined by (1) the size of the service area, (2) the number of stations in the service area, and (3) the density of the bus stops in the service area. The two sizes of the service area are square areas of 3×3 miles^2 and 5×5 miles^2. They are labeled as small and large problems respectively and 10 such problems are generated for each size. The number of stations in the service area is set to 4 stations for the small problems and 7 stations for the large problems. For the density of the bus stops in the service area, it is set between 4 to 6 stops/miles^2. The bus stops are randomly and uniformly located in the service area.

The problem structure is determined by three factors: (1) the shape of the station network, (2) the location of the destination station, and (3) the grouping of the bus stops. Three types of station networks are taken into consideration: a line network, a junction network and a crossing network. For the destination station, it is located either at the central or at the peripheral of the service area. The grouping of the bus stops are either clustered or evenly distributed based on the geographical location of the service area. The schematic diagrams to illustrate the problem structures are shown in Figures 3.1(a), 3.1(b) and 3.1(c).

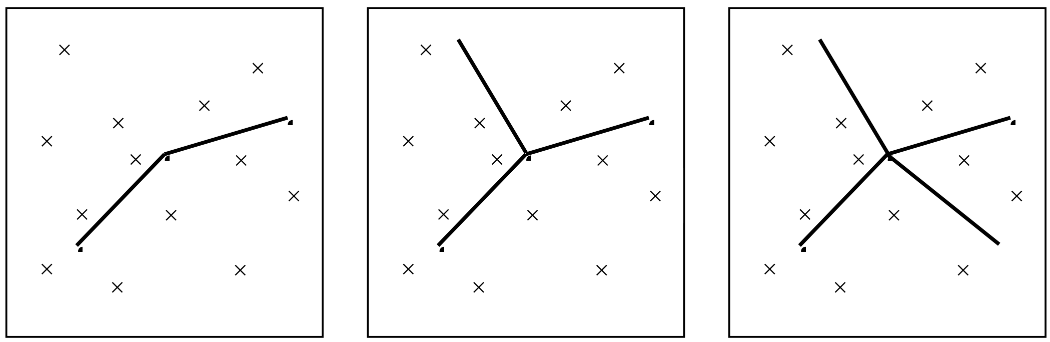

Figure 3.1(a): Illustration of the three types of station networks: line (left), junction (centre), and crossing (right)

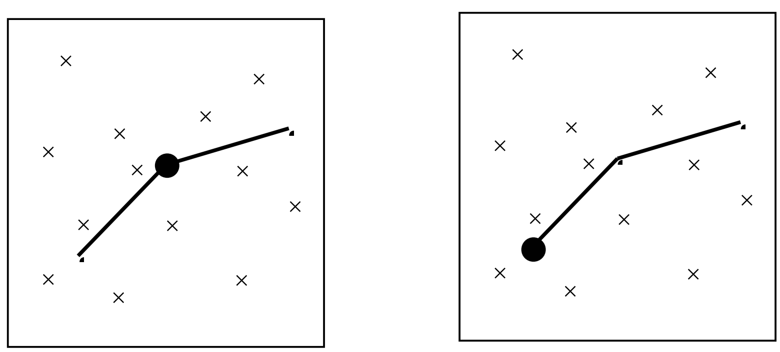

Figure 3.1(b): Illustration of the two locations of the destination station: central (left) and peripheral (right)

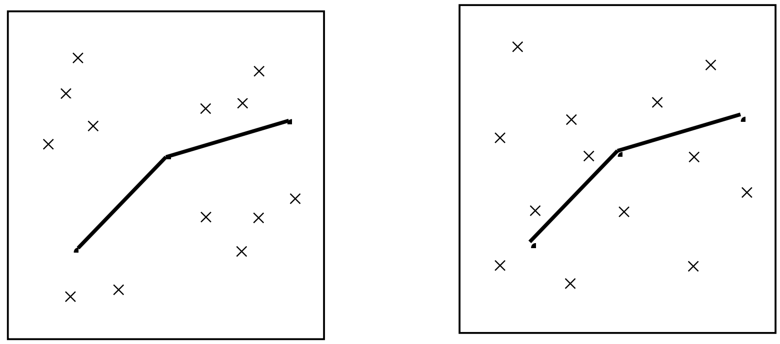

Figure 3.1(c): Illustration of the grouping of the bus stops: clustered (left) and evenly distributed (right)

For all these 20 randomly generated problems, the demand at each bus stop is fixed at 150 passengers per hour. The values for the other parameters are the same as the base problem, except for the maximum allowable route lengths. The maximum allowable route lengths are 2.5 miles and 4 miles for the small and large problems respectively. The characteristics of these problems are summarized in Table 3.4.

Table 3.4: Summary of the test problems generated

Problem	Service area (A)	No. of stations	Stop density (B)	No. of stops (A×B)	Station network	Location of destination station	Grouping of bus stops
1	3×3	4	4	36	line	peripheral	even
2	3×3	4	4	36	line	peripheral	clustered
3	3×3	4	4	36	line	peripheral	clustered
4	3×3	4	4	36	junction	central	even
5	3×3	4	4	36	junction	central	clustered
6	3×3	4	6	54	line	peripheral	even
7	3×3	4	6	54	line	peripheral	clustered
8	3×3	4	6	54	line	peripheral	clustered
9	3×3	4	6	54	junction	central	even
10	3×3	4	6	54	junction	central	clustered
11	5×5	7	4	100	line	peripheral	even
12	5×5	7	4	100	junction	peripheral	even
13	5×5	7	4	100	junction	central	clustered
14	5×5	7	4	100	crossing	peripheral	even
15	5×5	7	4	100	crossing	central	clustered
16	5×5	7	5	125	line	peripheral	even
17	5×5	7	5	125	junction	peripheral	even
18	5×5	7	5	125	junction	central	clustered
19	5×5	7	5	125	crossing	peripheral	even
20	5×5	7	5	125	crossing	central	clustered

3.5 Concluding Remarks

In this chapter, we have outlined our approach for solving the FBNDP. The initial solution is constructed on a stochastic basis together with the concept of delimiter to cut down search space to prevent too many bad selections that may result in a poor initial solution. The four metaheuristics which we will be proposing to solve the FBNDP are briefly introduced. The neighbourhood structure is also defined by three move types: the exchange move, the reduction-displacement move and the addition-displacement move. Besides using a base problem taken from the literature, we also randomly generated 20 test problems to further compare the performance of the metaheuristics proposed. These problems vary in several characteristics, such as the problem size and the problem structure.

Chapter 4 Applying Simulated Annealing to FBNDP

In this chapter, we present a general description of SA and its physical analogy, followed by the detailed procedures of how SA is applied to solve the FBNDP. Lastly, the computational results are presented.

4.1 General Description

SA was proposed by Kirkpatrick et al. (1983) for finding good solutions to combinatorial optimization problems. The main feature of SA is that it may avoid the problem of iterates getting trapped at a local minimum by probabilistically accepting a neighbourhood solution that is worse than the current one. This uphill movement makes it possible to move away from a local minimum and explore other feasible regions which may later lead to better solutions.

4.1.1 Physical analogy

SA is deduced from the physical annealing process of solids. In the annealing process, a solid is heated to a high temperature in which all molecules of the solid randomly arrange themselves into a liquid phase. The temperature of the melted solid is then gradually reduced to allow sufficient time for the molecules to arrange themselves naturally into a regular, stable and strong structure which is at the state of minimum energy (ground state). If the solid is cooled very quickly (rapid quenching), the solid is unable to reach thermal equilibrium for the intermediate temperature values, and irregularities and defects can be locked into the structure. The solid will then become brittle, with an energy level which is significantly higher.

In addition, for a solid at any temperature greater than zero, there exists a small probability that the solid will enter a state of higher energy (Metropolis et al., 1953). This implies that it is possible that the solid will leave the state of minimum energy for a new state where the energy is increased. But as the temperature is lowered, the probability that the energy level will increase suddenly is reduced. The probability *p* that a change in the state of the metal at some temperature *T* with initial energy level E_1 to some other state with energy level E_2 is given by:

$$p = e^{-\frac{(E_2 - E_1)}{\kappa T}} \quad \text{if } E_2 > E_1 \qquad (4.1)$$

where κ is the Boltzmann's constant.

4.1.2 The metaheuristic

This physical analogy can be adapted to combinatorial optimization. The feasible solutions can be considered as the states of the solid, the energies of the states as the objective function values, the minimum energy state as the optimal solution to the problem, and rapid quenching can be viewed as local optimization.

At any current solution, a neighbourhood solution is generated and its objective function value is calculated. The difference between the objective function values is denoted by Δ, which is analogous to the change in energy of the two states. If $\Delta < 0$, the neighbourhood solution is better (lower objective function value) and becomes the current solution. If $\Delta > 0$, the neighbourhood solution has a probability of $\exp(-\Delta/T)$ of being accepted as the new current solution, where T represents the current temperature.

Thus, unlike traditional descent methods where only those feasible solutions of lower objective function values are considered, SA accepts with certain probability feasible solutions which are worse than the current one. This prevents the possibility of being trapped in a local minimum as in the case with a descent algorithm. At the beginning of the process where the temperature is high, the acceptance probability is high so that every feasible neighbourhood solution has high chances of becoming the current solution. The search process is focused on diversification and exploration of new regions. As the temperature lowers, only solutions of good quality can have a higher probability of being accepted as the current solution. Thus, the search process has shifted to exploitation of the good solutions.

To apply SA, it is important to determine a good cooling schedule so that a high quality final solution can be obtained. In designing the cooling schedule, four parameters must be specified: the *initial temperature*, the *cooling function* with a *cooling rate* for lowering the temperature, the criterion for *thermal equilibrium* at each temperature step, and the *stopping rule*.

The *initial temperature* value is a trade-off between the attainability of all feasible solutions and the computational time and can be obtained experimentally. The *cooling function* defines the way the temperature is reduced after thermal equilibrium at each temperature is achieved. A simple cooling strategy is the geometric cooling rule such that the new temperature is a factor of the previous temperature. This factor is called the *cooling rate*, which is a constant less than but close to one. Faster cooling will lead to a local optimum similar to rapid quenching in the annealing process, while slower cooling leads to excessively long computation times and hence a compromise has to be made. The criterion to determine when *thermal equilibrium* at a given temperature is reached is necessary so that the number of newly generated feasible solutions for each temperature is large enough. Lastly, a *stopping rule* determines when the algorithm stops and is usually done by fixing a final temperature or a maximum number of total iterations.

The main disadvantage of SA is that it can produce good solutions at the expense of large computational times. This is mainly due to the tendency of getting caught in repetition of moves resulting in cycles. The stochastic nature of the algorithm also requires running the experiments many times in order to find the best solution.

4.2 Proposed Method

Neighbourhood structure

In SA, a neighbourhood solution is generated on a stochastic basis. As mentioned in Section 3.3, three types of moves are used when searching for a neighbourhood solution and in this SA implementation, each of them has an equal chance of being selected at every iteration. After a type of move has been chosen, the routes and the stops required to perform the move are selected at random. Because of the large search space involved, it is necessary to restrict the neighbourhood so that the number of bad moves selected can be reduced. The concept of *delimiter* is applied to the *exchange* move type whereby two stops from different routes are exchanged. This is similar to the way when it is used for constructing the initial solution as described in Section 3.1.

Firstly, calculate the distance between each stop i in route R_k and its nearest stop j belonging to another route R_l:

$$NN_i = \min_{j \in R_l, k \neq l} d_{ij}, \qquad i \in R_k \qquad (4.2)$$

Repeat the calculation for all stops and the *exchange delimiter* D_E is the maximum of the set of minimum distances previously calculated:

$$D_E = \max_i NN_i \qquad (4.3)$$

Thus, the distance between the stops of two different routes selected at random for the *exchange* move must be less than D_E, so that there is a higher potential of generating a move that will improve the quality of the neighbourhood solution.

Cooling function

The cooling function for the reduction of the temperature used is the simple geometric function. The temperature at iteration i is $T_i = cooling\_rate \times T_{i-1}$.

Thermal equilibrium

Thermal equilibrium at each temperature is achieved by fixing the number of moves to be performed at each temperature T to some predetermined value *max_moves*.

Stopping rule

The stopping rule is defined by setting a predetermined value *max_count*. Whenever thermal equilibrium at any given temperature is reached, the counter is incremented by one. However, the counter is reset to zero under two conditions: (1) when a new best solution is found, and (2) if the percentage of accepted moves for the current temperature is greater than a prespecified minimum percentage *min_percent*. When the counter reaches *max_count* value, the process is terminated.

SA procedure

Step 1: Generate an initial solution *s* and improve the solution by solving TSP for each route using the 2-opt procedure.

Step 2: Set the current temperature to the initial temperature $T = T_{init}$.

Step 3: Set *counter* = 0.

Step 4: Set the number of accepted moves = 0 and perform the following steps *max_moves* times at each temperature *T*:

4.1 Generate a solution *s'* by randomly selecting a move from the neighbourhood space of *s* and compute its total cost *TC(s')*. Calculate the difference $\Delta TC = TC(s') - TC(s)$. If $\Delta TC < 0$, accept the neighbourhood solution and go to Step 4.3.

4.2 Generate a random number $R \in U[0, 1]$ and calculate the acceptance probability $p = \exp(-\Delta TC/T)$. If $R < p$, accept the neighbourhood solution and go to Step 4.3. If $R \geq p$, go to Step 4.1.

4.3 Set $s = s'$ and increment the number of accepted moves by 1. Perform TSP for each route using 2-opt if it is the best solution so far.

Step 5: Decrease the current temperature using the cooling function $T = cooling\_rate \times T$.

Step 6: If a new best solution has been found or the percentage of accepted moves in the last *max_moves* is greater than some predetermined level *min_percent*, reset *counter* to zero and go to Step 4. Otherwise, increment *counter* by 1. If *counter* < *max_count* go to Step 4 and repeat the process.

Step 7: Return the best solution found.

4.3 Computational Results

The parameters are chosen experimentally to ensure a compromise between the running time and the solution quality. This is done by varying the parameter values over an estimated range and doing trial runs for all the problems generated. The values of *cooling_rate* and *max_moves* are related to the problem size (I is the number of stops and J is the number of stations) as follows: $cooling\_rate = \max[0.5,\ 0.99\,(1 - Ae^{-B(I+J)})]$, where $A = 0.57$ and $B = 0.0055$, and $max\_moves = max\ [500,\ 3 * (I + J)]$. Other parameters are found to be independent of the problem size and are set as: *max_count* = 100, initial temperature $T_{init} = 90$, and *min_percent* = 1%.

For each of the problems, a total of 20 runs are performed. In each run, an initial solution is generated randomly and then improving it by using SA. The average total costs, the best total costs and the average computational times are recorded. The results are summarized in Table 4.1 and Table 4.2.

Table 4.1: Summary of average and best total costs for SA

Problem	Average (out of 20 runs)	Best (out of 20 runs)
Base	6845.8	6519.8
1	4684.3	4461.5
2	4909.8	4581.1
3	4892.8	4685.6
4	4750.9	4413.3
5	4396.3	4195.4
6	7054.9	6749.1
7	7822.1	7576.3
8	7086.4	6773.3
9	6355.0	6113.2
10	6221.0	5951.0
11	17711.7	17297.1
12	17173.5	16635.6
13	14815.2	14548.5
14	16912.7	16325.9
15	14487.2	13938.2
16	21564.9	21134.2
17	20639.2	20000.9
18	18395.9	17943.6
19	21339.8	20299.5
20	17731.6	17270.0

Table 4.2: Summary of computational times in seconds for SA

Problem	Average (out of 20 runs)
Base	4.62
1	5.07
2	6.92
3	5.80
4	5.14
5	4.28
6	6.54
7	7.66
8	6.62
9	6.24
10	6.15
11	89.95
12	85.97
13	87.12
14	88.23
15	78.91
16	108.73
17	109.55
18	91.35
19	111.89
20	88.63

The computational results obtained for the base problem are an average total cost of 6845.8 and an average time of 4.62 seconds. The best total cost obtained in 20 experimental runs for the base problem is 6519.8. A comparative study of the SA against other algorithms is given in Chapter 8.

4.4 Concluding Remarks

In this chapter, we have discussed the implementation of SA for solving the FBNDP. This metaheuristic is able to avoid getting trapped at a local minimum by probabilistically accepting a neighbourhood solution that is worse than the current one. There are four important parameters defined to ensure a good cooling function: the initial temperature, the cooling rate for lowering the temperature, the number of neighbourhood moves to be performed at each temperature for thermal equilibrium, and the stopping rule by setting a maximum counter value. The neighbourhood structure is also defined using the concept of delimiter to cut down search space.

Chapter 5 Applying Tabu Search to FBNDP

In this chapter, we present a general description of TS which uses adaptable memory, followed by the detailed procedures of how TS is applied to solve the FBNDP. Lastly, the computational results are presented.

5.1 General Description

TS is a memory-based iterative search procedure introduced by Glover (1989) for solving combinatorial optimization problems. While other metaheuristics, such as SA and GA, are 'memoryless', TS makes use of flexible memory by exploiting its three memory components:

1. A short-term memory component, which is also called the *basic TS* algorithm.
2. An intermediate-term memory component, which is used for *intensifying* the search.
3. A long-term memory component, which is used for *diversifying* the search.

5.1.1 Basic TS

Basic TS simply uses the short-term memory component to store certain characteristics of recently visited solutions in a *tabu list* and classify these characteristics as *tabu* (forbidden or not allowed) for a prespecified number of iterations. By doing so, it can avoid returning to previously visited solutions temporarily and force the search into other regions. This can help to lower the possibility of cycling or repetition of moves that frequently occurs in SA algorithm.

When a neighbourhood solution is generated from a current solution, its move *attributes* are checked against the *tabu list*. If the move is not tabu, it becomes the current solution, even if its objective function becomes worse. This feature helps to prevent being entrapped in a local optimum. If the move is tabu, the *aspiration criterion* is checked. If it passes the aspiration criterion, then it becomes the current solution. The attributes of this current move are then stored in the tabu list, and the reversal of this move is not allowed for a tenure controlled by the *tabu list size*. Otherwise, the next best move not in the tabu list is selected and performed. Lastly, the search procedure is repeated until a *stopping criterion* is reached. The stopping criterion is usually a fixed number of iterations that is determined experimentally.

As mentioned earlier, TS has a memory component to store recently visited solutions and to prevent returning to these solutions for a short term. However, storing a list of complete solutions, even a small number of them, and comparing them with newly generated ones, takes up a lot of time effort and memory requirement. As such, only *attributes* of a move are stored in the tabu list to prevent the reversal of the move. Any change that results in the move from one solution to another can be an attribute of that move and there can be several attributes in order to define a move. For example, the 2-opt procedure for solving the TSP deletes two nonadjacent edges of a tour and adds back two edges to create a new tour. This move consists of at least two attributes, the added edges and the dropped edges.

When the attributes are stored in the *tabu list*, certain tabu restrictions are imposed. One example is, if a move contains attribute *m*, then the reverse attribute *m'* is forbidden. Take the 2-opt procedure for example: tabu restrictions include adding an edge which had been removed earlier, removing an edge which had been added earlier, or both.

A tabu restriction is imposed only when the attributes occurred within a limited number of iterations prior to the present iteration and is controlled by the *tabu list size*. A *recency-based* memory structure is used to remember how recently solutions have been visited, or how recently attributes of moves have been changed and classifying them as tabu. A queue (FIFO - First In, First Out) is usually used in the tabu list. Generally, the tabu list size is small, which corresponds to the short-term memory of TS. Usually, the more stringent the tabu restriction is, the smaller the tabu list size should be. Its value can be determined from experimental trials. Cycling occurs when the size is too small and deterioration of solution quality results when the size is too large and too many moves are forbidden.

Since only move attributes (not complete solutions) are stored in the tabu list, these tabu restrictions may also prevent the consideration of some unvisited good solutions. Thus, the *aspiration criterion* is introduced to override the tabu status of a recent move in the tabu list if the reversal of the move is sufficiently good. The most commonly used is the *best solution* aspiration criterion which overrides the tabu status when a move produces a solution better than the best solution obtained so far. If that is the case, the solution resulting from that move has satisfied the aspiration criterion and is accepted as the current solution.

5.1.2 Search intensification

The basic role of the intermediate-term memory component is to intensify the search, that is, to make the search focus on regions that are likely to be attractive and produce a better solution. Usually, a *frequency-based* memory structure is used and attributes that appear in accepted solutions with a certain high frequency over a larger number of iterations (greater than tabu list size) are recorded. These attributes are deemed to be attractive when seeking for a good solution and the search will be focused on preserving these attributes. For example, in the TSP, after some number of initial iterations, all edges not yet incorporated into any tours can be discarded. This can reduce the search space and also the computational time.

5.1.3 Search diversification

The goal of long-term memory component is to diversify the search. Instead of focusing the search into regions that contain previously found good solutions, it drives the search process into new regions that are different from those examined so far. A *frequency-based* memory structure is commonly used. Using the TSP as an example, one way to implement diversification is to penalize the edges that have occurred too frequently in the tours previously generated and focus the search on tours that do not include these edges. In this way, other unvisited regions can be explored in the hope that they can give better solutions.

5.2 Proposed Method

Neighbourhood structure

Instead of selecting neighbourhood solutions on a stochastic basis for the SA algorithm, in this TS implemented, a subset of the feasible neighbourhood solutions will be examined and the best among these will be selected. In order to reduce the neighbourhood space, the *candidate list* strategy is adopted. Here, for each stop, it will only be displaced next to its *n* nearest neighbours or exchange positions with its *n* nearest neighbours. It is assumed that displacing or exchanging a stop to/with neighbour stops that are further away is unlikely to form a good solution. Among these possible feasible moves, only the one that gives the best solution will be selected and checked for tabu status and aspiration criterion. If the best move selected is tabu and fails to satisfy the aspiration criterion, the next best move will be chosen. The size of nearest neighbours considered, *n_NB*, will be one of the parameters to be tested.

Move attributes and tabu list

The attributes stored in the tabu list are (1) the stop *i* that is being displaced or exchanged, (2) the stop *before i* after *i* has been displaced or exchanged, and (3) the stop *after i* after *i* has been displaced or exchanged. For the *addition-displacement* move type, since a new route is formed, the stops before and after *i* will be substituted by the station attached to stop *i* to indicate that the stop is only linked to the station only.

The tabu restrictions imposed will be to prevent the replacement of a stop in its previous position for some iterations. Hence, when displacement or exchange takes place, stop *i* cannot be re-inserted between attributes (2) and (3) (the stops *before* and *after i*) stored in the tabu list for that prespecified number of iterations.

Tabu list size

A FIFO queue is used in the tabu list. Every time the tabu list is updated, the last move performed is added to the end of the list and the oldest one on top of the list is removed. The tabu list size, *tabu_size*, will be determined experimentally.

Aspiration criterion

The best solution aspiration criterion is used, that is, the tabu status is overriden if a move produces a solution better than the best obtained so far during the search.

Stopping criterion

The stopping criterion is defined by setting a predetermined total number of iterations *max_Iter*.

Intensification

The intensification strategy proposed here to is exploit the elite solutions discovered in the search process. Whenever a best move chosen from the subset of neighbourhood results in a new best solution, the search will branch into intensification by choosing the next best move to obtain the current solution and this continues until *max_Iter* is reached. After that, it will return to the stage just before it is branched, using back the best move to obtain the current solution and the search proceeds as usual.

Basic TS procedure

Step 1: Generate an initial solution *s* and improve the solution by solving TSP for each route using the 2-opt procedure.

Step 2: Initialize the tabu list to empty set and set the aspiration level.

Step 3: Perform the following steps *max_Iter* times:

3.1 From the neighbourhood space of *s*, choose a best neighbour *s'* which is not tabu or has satisfied the aspiration criterion.

3.2 Set *s* = *s'* and increment the number of iterations by 1. Perform TSP for each route using 2-opt if it is the best solution so far. Update the tabu list and aspiration level.

Step 4: Return the best solution found.

TS with intensification procedure

Step 1: Generate an initial solution *s* and improve the solution by solving TSP for each route using the 2-opt procedure.

Step 2: Initialize the tabu list to empty set and set the aspiration level.

Step 3: Perform the following steps *max_Iter* times:

3.1 From the neighbourhood space of *s*, choose a best neighbour *s'* which is not tabu or has satisfied the aspiration criterion.

3.2 Set *s* = *s'* and increment the number of iterations by 1. If it is the best solution so far, do intensification. Perform TSP for each route using 2-opt for the best solution. Update the tabu list and aspiration level.

Step 4: Return the best solution found.

5.3 Computational Results

The parameters are chosen experimentally to ensure a compromise between the running time and the solution quality. This is done by varying the parameter values over an estimated range and doing trial runs for all the problems generated. The values of *n_NB* and *max_Iter* are related to the problem size (I is the number of stops and J is the number of stations) as follows: *n_NB* = 20% of I and *max_Iter* = $1.5 \times (I + J)$. The size of the tabu list, *tabu_size* is fixed at the value 10.

For each of the problems, a total of 20 runs are performed. In each run, an initial solution is generated randomly and then improving it by using basic TS and TS with intensification respectively. The average total costs, the best total costs and the average computational times are recorded. The results of basic TS are summarized in Table 5.1 and Table 5.2 and results of TS with intensification are summarized in Table 5.3 and Table 5.4.

Basic TS

Table 5.1: Summary of average and best total costs for Basic TS

Problem	Average (out of 20 runs)	Best (out of 20 runs)
Base	6549.7	6351.4
1	4528.1	4440.0
2	4675.0	4533.3
3	4694.8	4536.1
4	4444.3	4326.9
5	4158.2	4102.4
6	6855.7	6713.6
7	7669.4	7549.6
8	6842.3	6727.0
9	6066.3	5937.5
10	6033.2	5881.8
11	17639.1	17242.8
12	17040.1	16781.9
13	14763.4	14405.9
14	16765.7	16205.0
15	13646.3	13356.0
16	21630.6	21206.6
17	20368.4	19917.9
18	18017.8	17817.7
19	20857.8	20445.6
20	17145.6	16764.0

Table 5.2: Summary of computational times in seconds for Basic TS

Problem	Average (out of 20 runs)
Base	1.53
1	0.74
2	0.59
3	0.70
4	0.60
5	0.64
6	1.42
7	1.29
8	1.40
9	1.26
10	1.16
11	19.06
12	17.21
13	12.55
14	15.75
15	11.72
16	31.42
17	25.61
18	19.11
19	24.69
20	17.14

TS with intensification

Table 5.3: Summary of average and best total costs for TS with intensification

Problem	Average (out of 20 runs)	Best (out of 20 runs)
Base	6485.5	6338.5
1	4489.1	4431.6
2	4582.4	4467.2
3	4640.0	4520.8
4	4380.6	4258.7
5	4125.1	4084.7
6	6785.7	6660.3
7	7569.7	7459.2
8	6790.5	6633.1
9	5996.7	5856.7
10	5980.0	5854.1
11	17512.2	17201.7
12	16867.8	16670.9
13	14659.3	14405.9
14	16605.5	16158.6
15	13549.5	13304.5
16	21476.0	21092.2
17	20236.1	19842.8
18	17936.5	17628.4
19	20642.9	20279.5
20	17049.9	16681.1

Table 5.4: Summary of computational times in seconds for TS with intensification

Problem	Average (out of 20 runs)
Base	13.81
1	4.13
2	4.29
3	3.97
4	3.49
5	4.04
6	11.51
7	10.28
8	8.80
9	12.09
10	9.78
11	147.48
12	137.37
13	84.87
14	120.02
15	93.00
16	250.02
17	196.52
18	150.50
19	223.34
20	152.58

The computational results obtained for the base problem for basic TS are an average total cost of 6549.7, average time of 1.53 seconds and best total cost of 6351.4. For TS with intensification, the average total cost is 6485.5, average time is 13.81 seconds and best total cost is 6338.5. When compared to the results of SA, basic TS obtained better results for most of the problem instances, i.e. lower total costs, and in shorter times while TS with intensification further improves the solution quality. A comparative study of the basic TS and TS with intensification against other algorithms is given in Chapter 8.

5.4 Concluding Remarks

In this chapter, we use a basic TS, as well as combining an intensification strategy, to solve the FBNDP. Basic TS makes use of short-term memory structures to store certain attributes of recently visited solutions in tabu lists and forbid these attributes for some time to prevent the repetition of moves. The intensification strategy further exploits the elite solutions discovered during the search process. The attributes defined in our approach are the stops before and after the one being displaced or exchanged, after the move has been performed. Thus, this stop cannot be re-inserted between these two attributes within certain iterations. The stopping criterion is defined by setting the total number of iterations. For the neighbourhood structure, the candidate list strategy is adopted and each stop will only be displaced or exchanged positions with its nearer neighbours. The tabu list size, the total number of iterations, and the size of nearest neighbours are parameters to be obtained experimentally. The results have shown that basic TS obtained better results than SA in shorter times while TS with intensification further improves the solution quality.

Chapter 6 Applying Genetic Algorithm to FBNDP

In this chapter, we present a general description of GA and how it is adapted from the evolution theory, followed by the detailed procedures of how GA is applied to solve the FBNDP using a new crossover technique. Lastly, the computational results are presented.

6.1 General Description

GA is developed by Holland (1975), which is an intelligent search heuristic inspired by Darwin's theory about evolution. According to Darwin's evolution theory, only the best-fit individuals should survive and create new offsprings, whereas the least-fit individuals will be eliminated. The GA simulates this behaviour by maintaining a population of solutions with a fitness value associated with each solution. By means of some selection techniques and operators, a new population with better overall fitness is being reproduced and it replaces the old population. This cycle is repeated until a satisfactory solution is found.

6.1.1 Biological analogy

All living organisms consist of cells formed by the same set of chromosomes. A chromosome consists of genes. Each gene characterizes a trait, for example, colour of the eyes, and has its own position in the chromosome. During reproduction, genes from parents recombine or *crossover* in some ways to form a whole new chromosome. This newly created offspring may undergo *mutation* which changes some genes in the chromosome, mainly caused by errors in copying genes from parents. If the offspring inherits good characteristics from its parents, it will be fitter and it is more likely to survive and continue to reproduce and pass down its good characteristics to its next generation.

6.1.2 The metaheuristic

The best solution to a problem solved by GA is evolved. Unlike other heuristics, like SA and TS, which evaluate only one solution at a time, GA begins with a set of solutions called *population*; each solution corresponds to a chromosome. Solutions are selected according to their *fitness*, usually a function of the objective function value and a penalty being imposed if the solution is infeasible, to form new solutions (offsprings). The better the fitness value of the solutions, the more suitable they are for *reproduction* and hence, the more chances they will be selected to reproduce. The new offsprings created are placed in the new population and it is said that a new generation has evolved. This is repeated until a *terminating condition* (e.g. number of generations) is satisfied. This is motivated by the hope that the new generation will be better than the old one and a best solution can be found.

Representation

GA does not use much knowledge about the problem to be optimized and does not deal directly with the parameters of the problem. It works with codes which represent the parameters. Thus, the first step to solve a problem with GA is to define a way to represent the problem. There are generally two types of *representation* of a chromosome (solution) to a problem: binary or permutation. The type of representation to be chosen depends mainly on the type of the problem to be solved.

In binary encoding, every chromosome consists of a binary string of 0 or 1. Each bit in this string can represent some characteristics of the solution. An example of a problem that can be best represented by binary encoding is the Knapsack problem. The knapsack problem has a knapsack with a given capacity and a set of things with given values and sizes. The objective of the problem is to select which things to be put in the knapsack in order to maximize the value of things in the knapsack, but not exceeding the knapsack capacity. When encoded, each bit represents if the corresponding thing is in the knapsack.

Chromosome: 1 0 1 1 0 0 1 0

Permutation encoding is more commonly used in ordering problems, such as the TSP. Every chromosome is a string of integers, which represents numbers in a sequence. For the TSP, the chromosome represents the order of the cities the salesman will visit.

Chromosome: 1 5 3 2 6 4 7 8

Selection

The next step is to determine the criterion for selecting parents from a population for reproduction. This can be done in many ways, but the main idea is to select the parents with better fitness values (in the hope that the better parents will produce better offsprings). One example of the *selection procedure* is the Roulette Wheel selection. Fitter individuals occupy larger sectors on the wheel and parents are selected by a random spinning of the wheel. The better the chromosomes are, the more chances they will be selected.

Reproduction

Reproduction is the most important part of GA. It serves the purpose of combining the useful traits from the parent chromosomes and passing them on to the offsprings. The reproduction of GA consists of two operators, *crossover* and *mutation*. The performance of GA is highly influenced by these two operators and efficient implementation of these operators is largely responsible for good performance. Depending on the encoding of the problem, the crossover and mutation operators differ.

Crossover selects genes from parent chromosomes and recombines them to create a new offspring in the hope that the new chromosomes will have good parts of the old chromosomes and the new chromosomes will be better. For the binary encoding, a single-point crossover is performed by selecting one crossover point and swapping the binary strings after the crossover point for both parents.

Parent 1:	1 1 0 0 1 | **0 1 1**
Parent 2:	1 1 0 1 1 | **1 1 1**
Offspring 1:	1 1 0 0 1 | **1 1 1**
Offspring 2:	1 1 0 1 1 | **0 1 1**

Other types of crossover for binary encoding include the double-point crossover and uniform crossover.

For permutation encoded strings, where each integer gene appears only once in any chromosome, crossover methods for binary chromosomes cannot be applied because invalid offsprings with duplicated genes in one string will be produced. To prevent such invalid offsprings from being reproduced, other crossover operators such as PMX and ERX are used.

Partially mapped crossover (PMX) builds an offspring by choosing two cut points at random. For example:

Parent 1: 1 2 3 | 4 5 6 7 | 8 9

Parent 2: 4 5 2 | 1 8 7 6 | 9 3

First, the cut-out section of the two parents are swapped.

Offspring 1: x x x | 1 8 7 6 | x x

Offspring 2: x x x | 4 5 6 7 | x x

The next step is a series of swapping operations to be performed on the two parents. For each pair of genes at the same position from the two parents between the two cut points (i.e. {4, 1}, {5, 8}, {6, 7}, {7, 6}), their positions are swapped and the following offsprings are obtained.

Offspring 1: 4 2 3 | 1 8 7 6 | 5 9

Offspring 2: 1 8 2 | 4 5 6 7 | 9 3

Edge recombination crossover (ERX) builds an offspring by preserving more than 95% of the edges from the parents. The general idea is that an offspring should be built exclusively from the edges present in both parents. This is done with the help of the edge list created from both parents. The edge list provides, for each city, all the other cities connected to itself in at least one of the parents.

For example:

Parent 1: 1 2 3 4 5 6 7 8 9

Parent 2: 4 1 2 8 7 6 9 3 5

The edge lists for city 1 are cities 9, 2 and 4, for city 2 are cities 1, 3 and 8, and so on.

Select an initial city from one of the parents. Assume city 1 is selected, which is directly connected with three other cities: 9, 2, and 4. The next city selected is the one with the smallest number of unused edges in the edge list. Cities 4 and 2 have three edges and city 9 has four. Therefore, a random choice is made between cities 4 and 2. Assuming city 4 is selected, the candidates for the next city are now 3 and 5 (1 is already in the tour). City 5 is selected since it has only three edges as opposed to the four edges of city 3. Continuing with this procedure, the new offspring will be:

Offspring: 1 4 5 6 7 8 2 3 9

Such selection increases the chance that the completed tour has most edges selected from the parents. One problem that occurs with edge recombination is that cities are often left without a continuing edge and a new edge has to be reintroduced.

After a crossover is performed, *mutation* takes place. This is a form of diversifying the search by forcing the algorithm to search new areas. Mutation changes the new offspring randomly. The mutation depends on the encoding as well. For example, in binary encoding, selected bits are inverted from 1 to 0 or from 0 to 1.

Old offspring: 1 **1** 0 0 1 0 0 1

Mutated offspring: 1 **0** 0 0 1 0 0 1

For permutation encoding, mutation could be performed by exchanging the positions of two genes randomly.

Old offspring: 1 **2** 3 4 5 6 7 **8**

Mutated offspring: 1 **8** 3 4 5 6 7 **2**

Also, when creating a new population by crossover and mutation, there is a high chance that the best chromosome will be lost. Therefore, the idea of *elitism* is being introduced. This means that at least one best solution is copied without any changes to the new population, so the best solution found can survive to the next generation. Elitism can increase performance of GA very rapidly because it prevents losing the best-found solution.

Parameters of GA

The two most important parameters of GA are the *crossover probability* and the *mutation probability*. The crossover probability says how often will crossover be performed. If there is no crossover, the offspring is an exact copy of the parent. If there is a crossover, the offspring is made from parts of the parents' chromosomes. If crossover probability is 100%, then all offsprings are made by crossover. The mutation probability says how often the parts of the chromosome will be mutated. If there is no mutation, offspring is taken after crossover without any change. If mutation is performed, part of the chromosome is changed. If mutation probability is 100**%**, then all chromosomes are changed. The mutation probability should be kept low because GA would have become random search if mutation occurs too frequently.

There are also some other parameters of GA. The population size determines how many chromosomes (solutions) are in a population (in one generation). If there are too few chromosomes, GA has a few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA is slowed down. Lastly, the terminating condition may be based on a fixed number of generations.

6.2 Proposed Method

Representation

The representation of a FBNDP solution used here is an integer string of variable length, depending on the number of routes formed. Each string consists of several substrings, each representing one route in the solution. Each substring is a sequence of stops in a route ended by the station that these stops are attached to.

For example, assuming that 1 to 10 are stop indices and 11 to 13 are station indices, a solution in this form:

Route 1: 1 2 3 4 **11**

Route 2: 5 6 7 **12**

Route 3: 8 9 10 **13**

is represented as

1 2 3 4 **11** 5 6 7 **12** 8 9 10 **13**

This representation is unique and one chromosome can only be decoded to one solution. Decoding the chromosome into route configurations is very simple because every station in the string indicates the end of each route. By coding the solution in this way, the original route configurations can always be retrieved after decoding with no ambiguity.

Initial population and Fitness

Initial solutions in the population are generated at random, similar to SA and TS. The size of the population, *pop_size* is predetermined. Fitness of a solution is measured by its objective function value. The penalty function is not included in this research because feasible solutions are always ensured during the reproduction process.

Selection

After generating the initial population, a tournament selection mechanism, adopted from Tan et al. (2001a and 2001b), is used to select parents for reproduction. In this tournament selection, the population of n solutions is duplicated to get two identical copies. These two populations are arbitrarily ranked. For population P_1, each pair of adjacent chromosomes (with indices $2i$ and $2i + 1$) is compared. The one with smaller fitness value (lower objective function) qualifies to be a potential parent. After comparing all the $n/2$ pairs in P_1, $n/2$ 'fathers' f_i, are obtained. This process is repeated for population P_2 to get $n/2$ 'mothers' m_i. Subsequently, f_i and m_i are mated for reproduction in the next stage. This selection scheme allows average quality solutions to have some chances of being selected too, provided that they are paired up with solutions that are even worse.

Reproduction

In order to ensure the validity of the offsprings produced by crossover, as well as to preserve the edges in the parents, the following crossover technique is used to solve the FBNDP. Starting from the first gene of the parent string, similar to ERX, the edges next to this stop are listed. The shortest edge between the current stop and its edges is chosen as the next gene in the string. The process is continued until the feasibility of the current route is violated. When this occurs, the next stop in the parent string that has yet to be included in the offspring chromosome will be selected as the first stop of the next route. Also, when there are no more unvisited edges to be chosen, the current route is terminated and the same procedure is continued for the next route.

Using the following example:

Parent 1: 1 2 3 4 **11** 5 6 7 **12** 8 9 10 **13**

Parent 2: 10 7 4 **11** 1 3 **12** 5 9 **12** 2 6 8 **13**

Starting from the first gene in parent 1, the first gene in offspring 1 is chosen to be stop 1. Next, the edges next to stop 1 are stops 2 and 3. Assuming that $d_{13} < d_{12}$, stop 3 is chosen as the next gene in the offspring. Similarly, the edges next to stop 3 are stops 2 and 4 (stop 1 has already been included). Assuming that $d_{34} < d_{32}$, stop 4 is then chosen as the next gene. The process is continued until the feasibility of the current route is violated. Then, station 11 which is attached to the first stop in this route (stop 1) is put next on the string. Scanning through the parent 1 string, the next unvisited stop is stop 2 and it becomes the first stop of the next route and the process continues.

Offspring 1: 1 3 4 **11** 2 … **11** …

The advantage of this crossover method is that, besides being able to preserve most of the edges in the parent chromosomes, it can also inherit the best qualities from the parents as well.

Mutation is performed by simply swapping any two stops in the string at random. The crossover probability, P_c, and mutation probability, P_m, will be determined experimentally.

Elitism

To restore some of the good solutions in the parent population, the worst 4% chromosomes in the offspring are substituted with the best 4% in the parents (Tan et al., 2001a and 2001b).

Terminating condition

Terminating condition is based on a fixed number of generations, *n_gen*.

GA procedure

Step 1: Generate random initial population of size, *pop_size,* with each solution improved by solving TSP for each route using the 2-opt procedure.

Step 2: Evaluate the fitness value in terms of the objective function for each chromosome in the population.

Step 3: Create a new population by repeating the following steps until it is completely generated:

3.1 [Selection] Select two parent chromosomes from a population by tournament selection.

3.2 [Crossover] With a crossover probability P_c, perform crossover on the parents to form a new offspring. If no crossover was performed, the offspring is an exact copy of the parent.

3.3 [Mutation] With a mutation probability P_m, mutate new offspring at random.

3.4 [Accepting] Place the new offspring in a new population.

3.5 [Recovery] Replace the worse 4% chromosomes in the new population with the best 4% in the parent's population.

Step 4: Update the old population with the newly generated population.

Step 5: If the total number of generations, *n_gen*, is reached, perform TSP for each route using 2-opt to improve the solutions in the current population. Else, go to step 2.

Step 6: Return the best solution found.

6.3 Computational Results

The parameters are chosen experimentally to ensure a compromise between the running time and the solution quality. This is done by varying the parameter values over an estimated range and doing trial runs for all the problems generated. The number of generations, *n_gen*, is related to the problem size (I is the number of stops and J is the number of stations), where $n\_gen = 10 \times (I + J)$. The population size, *pop_size*, is set at 100. Other parameters are: the crossover probability $P_c = 0.8$ and the mutation probability $P_m = 0.06$.

For each of the problems, a total of 20 runs are performed. In each run, an initial population is generated randomly and then improving it by using GA. The average total costs, the best total costs and the average computational times are recorded. These results are summarized in Table 6.1 and Table 6.2.

Table 6.1: Summary of average and best total costs for GA

Problem	Average (out of 20 runs)	Best (out of 20 runs)
Base	6519.4	6412.5
1	4521.4	4447.2
2	4639.9	4546.3
3	4700.7	4576.4
4	4467.0	4380.5
5	4118.3	4111.2
6	6910.3	6785.8
7	7621.6	7599.2
8	6792.5	6706.6
9	6045.6	5992.5
10	5922.7	5903.9
11	17625.2	17502.9
12	16848.0	16767.9
13	14727.7	14611.0
14	16621.7	16547.5
15	13719.0	13683.9
16	21402.8	21315.9
17	20806.8	20634.4
18	18307.2	18178.2
19	21087.1	20963.8
20	17183.5	17065.2

Table 6.2: Summary of computational times in seconds for GA

Problem	Average (out of 20 runs)
Base	10.10
1	5.48
2	4.67
3	5.78
4	5.69
5	4.84
6	9.53
7	8.66
8	7.71
9	10.28
10	9.33
11	127.47
12	105.59
13	76.83
14	127.32
15	87.39
16	167.42
17	148.88
18	127.82
19	163.25
20	144.17

The computational results obtained for the base problem for GA are an average total cost of 6519.4, average time of 10.1 seconds and best total cost of 6412.5. The performance of GA is comparable to basic TS and GA achieves better results than SA. However, it is not as good as TS with intensification. The computational times are slightly shorter than that for TS with intensification. A comparative study of the GA against other algorithms is given in Chapter 8.

6.4 Concluding Remarks

In this chapter, GA is applied to solve the FBNDP. GA is inspired by Darwin's theory about evolution that only the fittest individuals should survive and create new offsprings. In our implementation, starting from an initial population of solutions, new and better generation of solutions are reproduced by going through the selection, crossover and mutation processes. Parent solutions are selected using the tournament selection technique. A new crossover operator is designed which is able to preserve most of the edges in the parent solutions, as well as able to inherit the best qualities from the parents, while ensuring the validity of the new offsprings produced. Mutation is performed by swapping any two stops in the parent chromosome at random. Elitism is also used to restore some of the good solutions in the parent population. The parameters in this approach are the crossover probability, the mutation probability, the size of a population and the number of generations as the terminating condition. The results have shown that GA achieves better results than SA, but it is not as good as TS with intensification.

Chapter 7 Applying Ant Colony Optimization to FBNDP

In this chapter, we present a general description of ACO and how it is inspired from the behaviour of ants, followed by the developments from the original Ant System to its recent extensions, and the detailed procedures of how ACO is applied to solve the FBNDP. Lastly, the computational results are presented.

7.1 General Description

ACO is a recently proposed metaheuristic approach for solving combinatorial optimization problems. ACO is inspired by the behaviour of real ants which lay pheromone trails on the paths they have travelled and use pheromones as a communication medium. Despite being a rather recent metaheuristic, ACO algorithms have already been applied to a large number of different combinatorial optimization problems, such as the vehicle routing problems.

7.1.1 Biological analogy

In many ant species, individual ants may deposit a pheromone (a particular chemical that ants can smell) on the ground while walking. By depositing pheromone, they create a trail that can be used to mark the path from their nest to food sources and back. By sensing pheromone trails left on the ground, foraging ants can follow the path to food discovered by other ants.

In one of the experiments done by researchers on ant behaviour, a bridge of two branches, one longer than the other, was used to connect a nest of ants, and a food source to study their pheromone trail laying and following behaviour. In this experiment, at the start, the ants were left free to move between the nest and the food source. As there was no pheromone deposited on the two branches initially, the ants had no preference for any of the two branches. Therefore, on average, half of the ants chose the short branch and the other half the long branch. However, because one branch was shorter than the other, the ants choosing the short branch were faster to reach the food and return back to the nest. This higher frequency of travelling on the shorter branch resulted in pheromone to accumulate faster and leaving a higher level of pheromone on the shorter branch. Eventually, it would be used by the majority of the ants. The outcome that most of the ants ended up using the shorter branch shows that they are also capable of exploiting pheromone trails to choose the shortest path among the available paths leading to the food.

7.1.2 The metaheuristic

In analogy to the biological example, ACO is based on the indirect communication of a colony of simple agents, called (artificial) ants, mediated by (artificial) pheromone trails.

Ant activity

An individual ant is capable of probabilistically constructing a feasible solution on its own. It has a memory that stores information about the path it has followed so far. Usually, it is assigned a start state of either a single component or an empty sequence, and iteratively adding components to its partial solution, while ensuring its feasibility (by implementing the constraints), until a complete solution is obtained.

When selecting components to be added, a probabilistic decision rule that makes use of (i) pheromone trails, and (ii) heuristic information is applied. Connections have associated pheromone trails τ that encode a long-term memory about how an ant searches. Heuristic values η represent a priori information about the problem, which are usually estimates of the cost of extending the current partial solution. Once a solution is built, the ant evaluates the solution and deposits pheromone trails, proportional to the quality of the solution produced, on the components or connections it has used. This pheromone information will change dynamically at run-time to reflect the ants' acquired search experience and direct the search of the future ants.

Ants move concurrently and independently and each ant is complex enough to find a (probably poor) solution to the problem under consideration. Typically, good quality solutions emerge as the result of the collective interaction among the ants through the pheromone trail values.

Trail update

Besides ants' activity, an ACO algorithm includes two more procedures: pheromone trail evaporation and daemon actions. Pheromone evaporation is the process in which the pheromone trail intensity on the components decreases over time. From a practical point of view, pheromone evaporation is needed to avoid too rapid convergence of the algorithm towards a sub-optimal region. It implements a useful form of forgetting, favouring the exploration of new areas of the search space. Daemon actions can be used to implement centralized actions that cannot be performed by single ants, such as the collection of global information from all the ants used and then deciding whether it is useful or not to deposit additional pheromone to bias the search process. A practical example is to choose to deposit extra pheromone on the components used by the ant that has built the best solution.

ACO in pseudo-code

Initialize
For a specified number of iterations do
    For ant k = 1 to m do
- *Construct a feasible solution iteratively using the probabilistic decision rule based on pheromone trails and heuristic information*
- *Calculate the objective value of the solution generated by ant k*

    End
    Update the pheromone trail levels
End

7.1.3 Ant System (AS)

The first ACO algorithm proposed in the literature was Ant System (AS) by Dorigo (1996) and was first applied to the TSP. In TSP, each ant is initially put on a randomly chosen city and has a memory which stores the partial solution it has constructed so far (initially the memory contains only the start city). Starting from its start city, an ant iteratively moves from city to city until the tour is completed. The probability that a city is selected is based on the intensity of the pheromone trail level leading to this city and the proximity of this city to the current city. When being at city i, an ant k chooses to go to a still unvisited city j with a probability given by

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{h \in \Omega} [\tau_{ih}]^{\alpha} \cdot [\eta_{ih}]^{\beta}} \qquad \text{if } j \in \Omega \qquad (7.1)$$

where

d_{ij}	distance between cities *i* and *j*
τ_{ij}	intensity of pheromone trail between cities *i* and *j*
$\eta_{ij} = 1/d_{ij}$	visibility of city *j* from city *i* (a priori available heuristic information)
α	parameter to regulate the influence of pheromone trail, τ_{ij}
β	parameter to regulate the influence of heuristic information, η_{ij}
Ω	feasible neighbourhood of ant *k*, that is, the set of cities which ant *k* has not yet visited

Parameters α and β have the following influence on the algorithm behaviour. If $\alpha = 0$, the selection probabilities are proportional to $[\eta_{ij}]^{\beta}$ and the closest cities will more likely be selected. In this case, AS corresponds to a classical stochastic greedy algorithm. If $\beta = 0$, only pheromone amplification is at work. This will lead to the rapid emergence of a stagnation situation and generating of tours which are suboptimal. (Search stagnation is the situation where all the ants follow the same path and construct the same solution.) The larger the value of α, the stronger the exploration of the search experience. Hence, varying α could be used to shift the search from exploration to exploitation and vice versa.

The solution construction ends after each ant has independently completed a tour. Next, the pheromone trails are updated. By analogy to nature, part of the pheromone evaporates, and this is done by lowering the existing pheromone trails by a constant factor, ρ. The parameter ρ is used to avoid unlimited accumulation of the pheromone trails and enables the algorithm to "forget" previous bad decisions. On edges which are not chosen by the ants, the associated pheromone strength will decrease exponentially with the number of iterations. If ρ is too close to one, most of the global information contained in the trail levels evaporate immediately and learning does not take place. If ρ is too close to zero, there is a danger of early convergence of the algorithm.

Following that, each ant will deposit pheromone on the edges that belong to its tour. The updating of trail level τ_{ij} is therefore,

$$\tau_{ij}^{new} = (1-\rho)\cdot\tau_{ij}^{old} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{7.2}$$

and $$\Delta\tau_{ij}^{k} = \begin{cases} Q/L^{k} & \textit{if edge } (i,j) \textit{ is used by ant } k \\ 0 & \textit{otherwise} \end{cases}$$

where

$\rho \in [0, 1]$ pheromone trail evaporation rate

m number of ants

$\Delta\tau_{ij}^{k}$ amount of pheromone ant k deposits on the edges (i, j)

Q quantity of pheromone laid by an ant per iteration

L^{k} length of the kth ant's tour

By equation (7.2), the amount of pheromone deposited by each ant is a function of the tour quality. The shorter the ant's tour is, the more pheromone is received by edges belonging to the tour. In general, edges which are used by many ants and which are contained in shorter tours will receive more pheromone and therefore are also more likely to be chosen in future iterations of the algorithm.

Although AS could not prove to be competitive with state-of-the-art algorithms specifically designed for the TSP when attacking large instances, it has stimulated further research on algorithmic variants to improve its performance.

7.1.4 Ant System and its extensions

A first improvement, called the elitist strategy (AS_{elite}), was introduced by Dorigo (1996). The idea of the elitist strategy in the context of the ant system is to give extra emphasis to the best path found so far after every iteration. In practice, each time the pheromone trails are updated, those belonging to the edges of the best tour get an additional amount of pheromone. The updating of trail levels is, therefore, done in this way:

$$\tau_{ij}^{new} = (1-\rho)\cdot\tau_{ij}^{old} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} + \Delta\tau_{ij}^{*} \qquad (7.3)$$

and $$\Delta\tau_{ij}^{*} = \begin{cases} \sigma\cdot Q/L^{*} & \textit{if edge } (i,j) \textit{ is part of the best solution found} \\ 0 & \textit{otherwise} \end{cases}$$

where

$\Delta\tau_{ij}^{*}$ increase of trail level on edge (i, j) caused by the elitist ants

σ number of elitist ants

L^{*} length of the best tour found

The edges of best tour are therefore reinforced with a quantity of $\Delta\tau_{ij}^{*}$ and this type of pheromone update is an example of daemon action mentioned earlier.

A subsequent improvement was the rank-based version of Ant System (AS_{rank}). AS_{rank} (Bullnheimer et al., 1999c) is in a sense an extension of the elitist strategy. After each tour construction phase, it sorts the ants according to the lengths of the tours they generated and the contribution of an ant to the pheromone trail level update is weighted according to the rank μ of the ant. In addition, only the μth best ants and the overall-best ant are allowed to deposit pheromone. The μth best ant of the colony contributes to the pheromone update with a weight given by (σ - μ), while the overall-best ant reinforces the pheromone trails with weight σ. The new updating of trail levels therefore becomes:

$$\tau_{ij}^{new} = (1-\rho)\cdot\tau_{ij}^{old} + \sum_{\mu=1}^{\sigma-1}\Delta\tau_{ij}^{\mu} + \Delta\tau_{ij}^{*} \tag{7.4}$$

and $$\Delta\tau_{ij}^{\mu} = \begin{cases} (\sigma-\mu)\cdot Q/L^{\mu} & \textit{if the } \mu\textit{th best ant travels on edge } (i,j) \\ 0 & \textit{otherwise} \end{cases}$$

where

μ ranking index

$\Delta\tau_{ij}^{\mu}$ increase of trail level on edge (i, j) caused by the μth best ant

L^{μ} tour length of the μth best ant

7.2 Proposed Method

Since FBNDP and VRP are closely related, the proposed method will be highly influenced by the application of the ant system to VRP (Bullnheimer et al., 1999a, 1999b and 1999c).

Ant activity

Firstly, each ant constructs a solution on its own. The procedure is as follows: an ant is assigned to a stop initially. Due to the multi-station nature of the problem, a decision has to be made to determine which station this stop has to link to. Here, a new probability is defined to measure how desirable it is to link stop *i* to station *st*:

$$p_{i,st}^{k} = \frac{[\tau_{i,st}]^{\alpha} \cdot [\eta_{i,st}]^{\beta}}{\sum_{H \in \Omega_{st}} [\tau_{i,H}]^{\alpha} \cdot [\eta_{i,H}]^{\beta}} \qquad \text{if } st \in \Omega_{st} \qquad (7.5)$$

where

d_i^{st} distance between stop *i* and station *st*

$\tau_{i,st}$ intensity of *pseudo*-pheromone trail between stop *i* and station *st* (the term 'pseudo' is used because stop *i* is not directly linked to station *st*, it simply means that stop *i* is being assigned to station *st*)

$\eta_{i,st} = 1/d_i^{st}$ visibility of station *st* from stop *i* (a priori available heuristic information)

α parameter to regulate the influence of pheromone trail, $\tau_{i,st}$

β parameter to regulate the influence of heuristic information, $\eta_{i,st}$

Ω_{st} the set of stations which stop *i* can be assigned to

After a station has been chosen according to the probability above, a route will be constructed by successively choosing stops to be visited, until the feasibility of the route becomes invalid. When this happens, the current route will be ended and the next unvisited stop will be selected and the process continues until all stops are visited. The probability of visiting stop *j* from stop *i* is similar to that for the TSP:

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{h \in \Omega} [\tau_{ih}]^{\alpha} \cdot [\eta_{ih}]^{\beta}} \qquad \text{if } j \in \Omega \qquad (7.6)$$

However, in the VRP (and also FBNDP), not only is the relative location of two cities important, but also their relative location to the depot is essential. This is the so-called *savings*, proposed by Clarke and Wright (1964), which measures the favourability of linking stop *j* after stop *i* with respect to the station which stop *i* is assigned to. Thus,

$$\eta_{ij} = s_{ij}^{st} = d_{i}^{st} + d_{j}^{st} - d_{ij} \qquad (7.7)$$

However, Tillman and Cain (1972) discovered that this savings equation is only applicable to the case of a single station. With their concept of 'modified distance' for the case of multiple stations, the modified distance for each stop *i* from each station *st* is computed as follows:

$$\tilde{d}_{i}^{st} = \min_{m} d_{i}^{m} - (d_{i}^{st} - \min_{m} d_{i}^{m}) \qquad (7.8)$$

where $\min_{m} d_{i}^{m}$ is the distance between stop *i* and its nearest station *m*.

The modified savings from linking stops *i* and *j* to a route assigned to station *st* then becomes

$$\eta_{ij} = \tilde{s}_{ij}^{st} = \tilde{d}_{i}^{st} + \tilde{d}_{j}^{st} - d_{ij} \qquad (7.9)$$

Trail update

After all artificial ants have constructed a feasible solution, the pheromone trails are updated depending on the quality of their objective function value, the total cost TC^k. A combined elitist and ranking strategy is used and the trail intensities are updated as follows:

$$\tau_{ij}^{new} = (1-\rho)\cdot\tau_{ij}^{old} + \sum_{\mu=1}^{\sigma-1}\Delta\tau_{ij}^{\mu} + \Delta\tau_{ij}^{*} \qquad (7.10)$$

The first term calculates how much the existing pheromone trails evaporates and is controlled by ρ, the *trail evaporation rate*. The second term shows that the pheromone trail is increased by $\Delta\tau_{ij}^{\mu} = (\sigma-\mu)\cdot Q/TC^{\mu}$ if edge (*i*, *j*) is used by the μth best ant with total cost TC^{μ} (zero otherwise), where μ is the rank of the ant according to solution quality. For the third term, for all σ elitist ants, if the edges (*i*, *j*) travelled by them also belong to the best solution found so far (total cost TC^{*}), the trail intensity is increased by an amount $\Delta\tau_{ij}^{*} = \sigma\cdot Q/TC^{*}$ (zero otherwise).

Candidate list

As with the similar difficulty encountered by any heuristics when solving a problem with big-sized neighbourhood, the solution construction by an ant without any restriction is very time consuming and the probability that many ants visit the same state is very small. Hence, the candidate list strategy, adapted from TS, is also applied here. During the construction, when an ant tries to move from one stop to another, only its *n_NB* nearest neighbours are considered. If the list of its nearest stops have all already been visited, the current route is ended and a new route has to be constructed.

Number of ants and their initial placement

Studies for VRP (Bullnheimer et al., 1999a) have shown that ACO yields very good results when the number of ants is set equal to the number of cities and each ant starts its tour from a city distinct from that of other ants. Thus, for the FBNDP, the number of ants is set equal to the number of stops and each ant is placed at each stop during initialization. The number of elitist ants σ is set experimentally.

Number of iterations

The number of iterations *max_Iter* is determined experimentally. In fact, the use of *m* ants that build *max_Iter* solutions each is equivalent to generating ($m \times max\_Iter$) initial solutions.

ACO procedure

Step 1: Initialize.

Step 2: For *max_Iter* iterations:

2.1 For each ant $k = 1$ to m, generate a new solution using equations (7.5) and (7.6) and improve the solution by solving TSP for each route using the 2-opt procedure.

2.2 Calculate the total cost for all solutions and rank them accordingly.

2.3 Update the pheromone trails using equation (7.10).

2.4 If a new best solution has been found, perform TSP for each route using 2-opt.

Step 3: Return the best solution found.

7.3 Computational Results

The parameters are chosen experimentally to ensure a compromise between the running time and the solution quality. This is done by varying the parameter values over an estimated range and doing trial runs for all the problems generated. The values of *max_Iter* and *n_NB* are related to the problem size (I is the number of stops and J is the number of stations) as follows: *max_Iter* = $1.5 \times (I + J)$ and *n_NB* = 20% of I. Other parameters are set as: $\alpha = 1$, $\beta = 5$, $\rho = 0.15$, $\sigma = 6$ and $Q = 100$.

For each of the problems, a total of 20 runs are performed and the average total costs, the best total costs and the average computational times are recorded. The results are summarized in Table 7.1 and Table 7.2.

Table 7.1: Summary of average and best total costs for ACO

Problem	Average (out of 20 runs)	Best (out of 20 runs)
Base	6716.5	6535.1
1	4646.5	4478.3
2	4867.4	4689.7
3	4791.5	4643.7
4	4840.4	4639.0
5	4403.8	4234.4
6	7041.6	6838.7
7	7861.2	7566.4
8	7007.7	6727.4
9	6226.1	5912.5
10	6185.5	5979.5
11	17816.1	17326.8
12	17292.7	16815.1
13	14876.6	14461.5
14	16933.6	16030.4
15	14454.2	14225.0
16	21675.5	21166.2
17	21485.8	20680.1
18	18417.8	18010.3
19	21597.4	21048.3
20	18131.1	17363.0

Table 7.2: Summary of computational times in seconds for ACO

Problem	Average (out of 20 runs)
Base	8.95
1	5.00
2	7.41
3	5.96
4	6.12
5	4.67
6	9.55
7	10.87
8	9.96
9	8.48
10	9.21
11	113.63
12	92.82
13	103.81
14	109.77
15	97.26
16	131.84
17	120.92
18	107.19
19	139.77
20	110.39

The computational results obtained for the base problem for ACO are an average total cost of 6716.5, average time of 8.95 seconds and best total cost of 6535.1. This result is comparable to SA and ACO is able to generate reasonably good solutions in short computational times. A comparative study of the ACO against other algorithms is given in Chapter 8.

7.4 Concluding Remarks

In this chapter, a very recently developed metaheuristic, ACO is used to solve the FBNDP. ACO is inspired by the behaviour of real ants which lay pheromone trails on the paths they have travelled to direct the search of the future ants. In our proposed method, each artificial ant constructs a feasible solution on its own using a probabilistic decision rule that makes use of pheromone trails and a priori information about the problem. Two probability functions are defined to calculate the desirability of linking each stop to its neighbouring stops and to the stations in the network. After all the ants have constructed their feasible solutions, the pheromone trails are updated depending on the quality of their objective function value. A combined elitist and ranking strategy is used and ants deposit pheromone on the edges it has used if it is one of the elitist ants or if the edges travelled by them also belong to the best solution found so far. The number of elitist ants and the total number of iterations are set experimentally. The results show that ACO is comparable to SA and is able to generate reasonably good solutions in short computational times.

Chapter 8 Analysis of Results

In this chapter, we analyze and compare the results of the metaheuristics that we have proposed in terms of their solution quality and computational efficiency. The results are also compared to those published in literature. We also check if the performances of these metaheuristics are problem-dependent.

8.1 Comparison of Results of Competing Metaheuristics

Let us first summarize the results we have obtained for the five methods we have attempted. The average total costs, the best total costs and the average computational times obtained for the base problem and the set of 20 randomly generated problems are shown in Table 8.1, Table 8.2 and Table 8.3 respectively.

Table 8.1: Comparison of average total costs

Problem	SA	Basic TS	TS with Intensification	GA	ACO
Base	6845.8	6549.7	6485.5	6519.4	6716.5
1	4684.3	4528.1	4489.1	4521.4	4646.5
2	4909.8	4675.0	4582.4	4639.9	4867.4
3	4892.8	4694.8	4640.0	4700.7	4791.5
4	4750.9	4444.3	4380.6	4467.0	4840.4
5	4396.3	4158.2	4125.1	4118.3	4403.8
6	7054.9	6855.7	6785.7	6910.3	7041.6
7	7822.1	7669.4	7569.7	7621.6	7861.2
8	7086.4	6842.3	6790.5	6792.5	7007.7
9	6355.0	6066.3	5996.7	6045.6	6226.1
10	6221.0	6033.2	5980.0	5922.7	6185.5
11	17711.7	17639.1	17512.2	17625.2	17816.1
12	17173.5	17040.1	16867.8	16848.0	17292.7
13	14815.2	14763.4	14659.3	14727.7	14876.6
14	16912.7	16765.7	16605.5	16621.7	16933.6
15	14487.2	13646.3	13549.5	13719.0	14454.2
16	21564.9	21630.6	21476.0	21402.8	21675.5
17	20639.2	20368.4	20236.1	20806.8	21485.8
18	18395.9	18017.8	17936.5	18307.2	18417.8
19	21339.8	20857.8	20642.9	21087.1	21597.4
20	17731.6	17145.6	17049.9	17183.5	18131.1

Table 8.2: Comparison of best total costs

Problem	SA	Basic TS	TS with Intensification	GA	ACO
Base	6519.8	6351.4	**6338.5**	6412.5	6535.1
1	4461.5	4440.0	**4431.6**	4447.2	4478.3
2	4581.1	4533.3	**4467.2**	4546.3	4689.7
3	4685.6	4536.1	**4520.8**	4576.4	4643.7
4	4413.3	4326.9	**4258.7**	4380.5	4639.0
5	4195.4	4102.4	**4084.7**	4111.2	4234.4
6	6749.1	6713.6	**6660.3**	6785.8	6838.7
7	7576.3	7549.6	**7459.2**	7599.2	7566.4
8	6773.3	6727.0	**6633.1**	6706.6	6727.4
9	6113.2	5937.5	**5856.7**	5992.5	5912.5
10	5951.0	5881.8	**5854.1**	5903.9	5979.5
11	17297.1	17242.8	**17201.7**	17502.9	17326.8
12	**16635.6**	16781.9	16670.9	16767.9	16815.1
13	14548.5	**14405.9**	14405.9	14611.0	14461.5
14	16325.9	16205.0	16158.6	16547.5	**16030.4**
15	13938.2	13356.0	**13304.5**	13683.9	14225.0
16	21134.2	21206.6	**21092.2**	21315.9	21166.2
17	20000.9	19917.9	**19842.8**	20634.4	20680.1
18	17943.6	17817.7	**17628.4**	18178.2	18010.3
19	20299.5	20445.6	**20279.5**	20963.8	21048.3
20	17270.0	16764.0	**16681.1**	17065.2	17363.0

Table 8.3: Comparison of average computational times in seconds

Problem	SA	Basic TS	TS with Intensification	GA	ACO
Base	4.62	1.53	13.81	10.10	8.95
1	5.07	0.74	4.13	5.48	5.00
2	6.92	0.59	4.29	4.67	7.41
3	5.80	0.70	3.97	5.78	5.96
4	5.14	0.60	3.49	5.69	6.12
5	4.28	0.64	4.04	4.84	4.67
6	6.54	1.42	11.51	9.53	9.55
7	7.66	1.29	10.28	8.66	10.87
8	6.62	1.40	8.80	7.71	9.96
9	6.24	1.26	12.09	10.28	8.48
10	6.15	1.16	9.78	9.33	9.21
11	89.95	19.06	147.48	127.47	113.63
12	85.97	17.21	137.37	105.59	92.82
13	87.12	12.55	84.87	76.83	103.81
14	88.23	15.75	120.02	127.32	109.77
15	78.91	11.72	93.00	87.39	97.26
16	108.73	31.42	250.02	167.42	131.84
17	109.55	25.61	196.52	148.88	120.92
18	91.35	19.11	150.50	127.82	107.19
19	111.89	24.69	223.34	163.25	139.77
20	88.63	17.14	152.58	144.17	110.39

In Table 8.4, we have also computed the average percentage deviation based on the average total costs from the best-known solution achieved. In Table 8.5, we further categorize the results according to the characteristics of the problems tested so as to check for any dominant methods for the different types of problems.

From these tables, we are able to summarize the comparison between these metaheuristics as follows:

1. In terms of average total cost, TS with intensification obtains the best results with only an average percentage deviation of 2.0% from the best-known solution. Basic TS and GA are the next best metaheuristics obtaining 3.0% and 3.1% deviation respectively, and are only about 1% worse than TS with intensification. Both are closely comparable. SA and ACO obtain 6.1% and 6.3% deviation respectively, and are approximately 3% worse than GA and basic TS, and 4% worse than TS with intensification. In general, SA and ACO are still able to give satisfactory results.

2. In terms of the best total cost, TS with intensification produces the best solutions in almost all the problems we have tested except for problems 12 and 14. For problem 12, SA gives the minimum total cost and for problem 14, ACO achieves the best result. For problem 13, basic TS obtains the best result and the intensification strategy does not further improve its solution quality.

3. In terms of the problem types, the performance of TS with intensification seems to be independent of the various characteristics of the problems such as the problem

size, the station network, the location of the destination station and the geographical distribution of the bus stops. It is able to outperform other metaheuristics regardless of the nature of the problem. This result shows the robustness of the intensification strategy. When comparing between GA and basic TS, no dominating heuristic is apparent. However, GA seems to be less effective when dealing with larger problem sizes. Between SA and ACO, ACO seems to favour problems with smaller sizes, single line station network, destination station located peripherally, and with evenly distributed bus stop locations. This result is encouraging because it is among the first attempts to implement ACO for route design problems and it shows ACO's competitiveness to the state-of-the-art heuristics like SA.

Table 8.4: Average percentage deviation from best solution obtained

Problem	Best known	SA (%)	Basic TS (%)	TS with Intensification (%)	GA (%)	ACO (%)
Base	6338.5	8.0	3.3	2.3	2.9	6.0
1	4431.6	5.7	2.2	1.3	2.0	4.9
2	4467.2	9.9	4.7	2.6	3.9	9.0
3	4520.8	8.2	3.9	2.6	4.0	6.0
4	4258.7	11.6	4.4	2.9	4.9	13.7
5	4084.7	7.6	1.8	1.0	1.8	7.8
6	6660.3	5.9	2.9	1.9	3.8	5.7
7	7459.2	4.9	2.8	1.5	2.2	5.4
8	6633.1	6.8	3.2	2.4	2.4	5.6
9	5856.7	8.5	3.6	2.4	3.2	6.3
10	5854.1	6.3	3.1	2.2	3.2	5.7
11	17201.7	3.0	2.5	1.8	2.5	3.6
12	16635.6	3.2	2.4	1.4	2.3	4.0
13	14405.9	2.8	2.5	1.8	2.2	3.3
14	16030.4	5.5	4.6	3.6	3.7	5.6
15	13304.5	8.9	2.6	1.8	3.1	8.6
16	21092.2	2.2	2.6	1.8	2.5	2.8
17	19842.8	4.0	2.6	2.0	4.9	8.3
18	17628.4	4.4	2.2	1.7	3.9	4.5
19	20279.5	5.2	2.9	1.8	4.0	6.5
20	16681.1	6.3	2.8	2.2	3.0	8.7
Average % deviation		6.1	3.0	2.0	3.1	6.3

Percentage deviation = (Average total cost – best known result) / best known result * 100%

Table 8.5: Comparison of average percentage deviation based on problem characteristics

Problem characteristics	SA	Basic TS	TS with Intensification	GA	ACO
Problem size					
▪ Small[1]	7.7	3.3	2.1	3.1	7.0
▪ Large[2]	4.6	2.8	2.0	3.2	5.3
Station network					
▪ Line[3]	7.1	2.9	1.8	2.8	6.6
▪ Junction	5.9	3.1	2.2	3.3	6.2
▪ Crossing	4.5	3.0	2.2	3.4	5.0
Destination station					
▪ Central	6.6	3.4	2.4	3.7	7.0
▪ Peripheral[3]	5.8	2.8	1.8	2.8	5.6
Bus stop distribution					
▪ Clustered	6.4	3.3	2.1	3.5	6.9
▪ Evenly distributed[3]	5.8	2.8	1.9	2.8	5.4

[1] Base problem and Problems 1 to 10
[2] Problems 11 to 20
[3] Classification of base problem

8.2 Comparison with Best-known Results

In this section, we compare the best solutions found for the base problem from our metaheuristics as well as those published from the literature.

The best solution obtained in this study is produced by TS with intensification, shown in Table 8.6 and Figure 8.1. A total cost of $6338.5/hr is achieved and 19 feeder bus routes are formed, with service frequencies ranging from 17 to 29 trips per hour (buses arriving at intervals of 2.1 to 3.5 minutes).

Table 8.6: Best solution obtained for base problem

Route number	Route structure						Route demand (passengers /hr)	Route length (miles)	Route frequency (trips/hr)
1	33	28	27	58			600	0.61	25.63
2	8	14	15	26	57		800	1.24	20.75
3	13	59					200	0.29	21.37
4	18	59					200	0.29	21.57
5	52	45	41	32	58		800	1.24	20.73
6	29	34	56				400	0.43	24.85
7	4	6	12	59			600	0.84	21.79
8	39	38	56				400	0.46	24.14
9	3	5	7	11	17	58	1000	1.35	22.25
10	16	19	23	30	56		800	1.27	20.53
11	44	49	51	46	42	56	1000	1.38	21.99
12	2	1	9	10	57		800	1.80	17.21
13	55	50	48	56			600	0.84	21.84
14	40	43	36	31	57		800	0.97	23.42
15	20	24	25	57			600	0.59	25.98
16	54	53	47	56			600	0.81	22.21
17	21	22	58				400	0.34	28.14
18	37	56					200	0.30	21.05
19	35	57					200	0.18	26.93

Total cost = $6338.5/hr

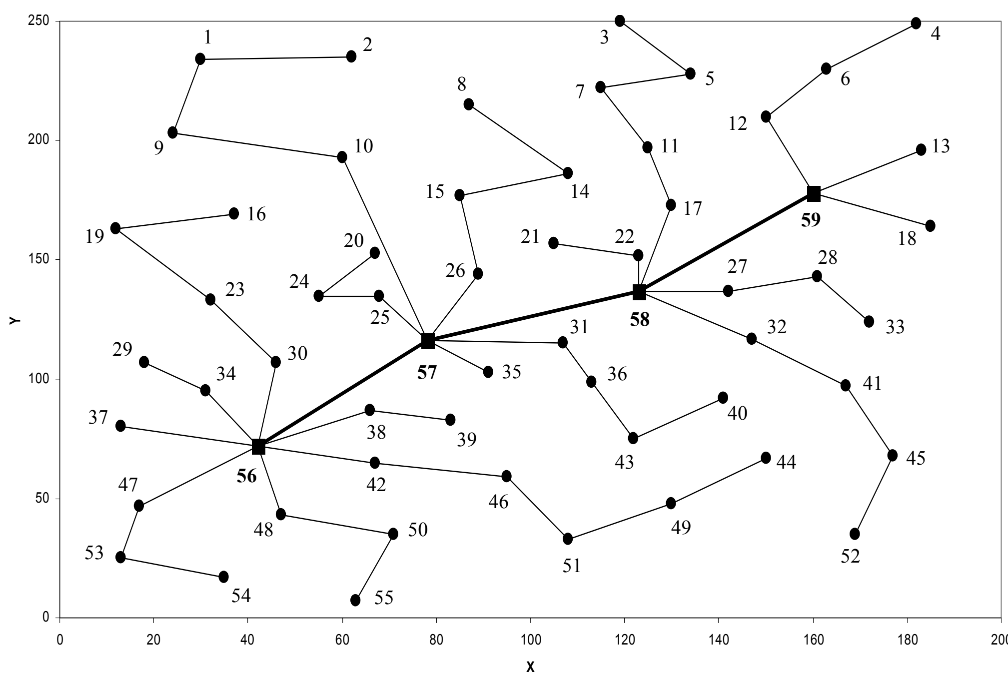

Figure 8.1: Best solution obtained for base problem

A comparison between the metaheuristics that we have attempted and the best-known results in literature for the base problem are shown in Table 8.7 below. As we can see, TS with intensification produces a new best solution.

Table 8.7: Comparison between our metaheuristics and the best-known results

Heuristic Approach	Total cost
Literature	
▪ Savings heuristic[1]	6846
▪ Tabu search[2]	6368
Our metaheuristics	
▪ SA	6519
▪ Basic TS	6351
▪ TS with Intensification	**6338**
▪ GA	6412
▪ ACO	6535

[1] From Kuah and Perl (1989)
[2] From Martins and Pato (1998)

8.3 Comparison of Computational Times

Table 8.8 below compares the average computational times recorded for the various metaheuristics again based on the characteristics of problems tested.

Table 8.8: Comparison of computational times based on problem characteristics

Problem characteristics	SA	Basic TS	TS with Intensification	GA	ACO
Problem size					
▪ Small[1]	5.9	1.0	7.8	7.5	7.8
▪ Large[2]	94.0	19.4	155.6	127.6	112.7
Station network					
▪ Line[3]	26.9	6.5	50.5	38.5	33.7
▪ Junction	49.8	9.6	72.7	63.9	58.8
▪ Crossing	91.9	17.3	147.2	130.5	114.3
Destination station					
▪ Central	46.0	8.0	63.8	58.3	55.9
▪ Peripheral[3]	49.0	10.9	87.0	68.6	59.0
Bus stop distribution					
▪ Clustered	38.3	6.6	52.2	47.7	46.7
▪ Evenly distributed[3]	56.5	12.7	101.8	80.1	67.9

[1] Base problem and Problems 1 to 10
[2] Problems 11 to 20
[3] Classification of base problem

As shown in the table, basic TS is the fastest algorithm as it is simple and it uses a candidate list strategy to restrict the neighbourhood to a reasonable size. With the intensification strategy, the computational time escalates. This result is expected because every current solution that achieves a new best solution is further intensified by searching its entire neighbourhood and this is time-consuming. However, we know that this strategy helps to improve the solution quality greatly and that the time taken is only a few minutes longer and hence acceptable. SA takes a longer time than basic TS. This is probably due to the reset of the counter whenever a new best solution is found or the percentage of accepted moves exceeds a minimum value. GA and ACO require similar amount of time. GA requires more time to perform the crossover operations because validity checks have to be done repeatedly to ensure that the maximum route length is not exceeded. ACO spends more time on updating the pheromone trails by checking for elitist ants and desirable edges. Lastly, no significant conclusion can be drawn based on the effect of the nature of the problem on the computational times. All the algorithms are coded in Visual C++ and were run on a Pentium 900 Mhz PC under Windows XP.

8.4 Concluding Remarks

In this chapter, a comparative study of the metaheuristics based on the base problem and the set of 20 randomly generated problems has been performed. The results obtained from our metaheuristics have also been compared with the best-known results in literature. In general, TS with intensification is the most effective metaheuristic, as it is generating better quality solutions and also obtaining the best solutions for most problems. However, its computational time is the longest. GA is closely comparable to basic TS, and they are slightly worse than TS with intensification. SA is fast and offers reasonably good solutions. ACO is able to give satisfactory results. In terms of relative computational times, basic TS is the fastest algorithm. GA and ACO require similar amount of time.

TS with intensification also produces a new best solution as compared to the literature. In terms of the problem types, the performance of TS with intensification is found to be independent of the various characteristics of the problem, showing the robustness of the intensification strategy. ACO is able to surpass some of the state-of-the-art heuristics like SA for certain types of problems. Lastly, no significant conclusion can be drawn based on the effect of the nature of the problem on the computational times.

Chapter 9 Summary and Conclusion

In this thesis, we have presented our methodology for solving the FBNDP. The FBNDP focuses on the design of a set of feeder bus routes and the determination of the operating frequency on each route, such that the objective function of the sum of operator and user costs is minimized. According to the literature review, the FBNDP belongs to the class of NP-hard, combinatorial problems that suffer from several forms of mathematical complexity and needs to be solved heuristically. Moreover, the literature survey also reveals that there has not been substantial work for both the analytic and network approaches to the feeder bus network design problem. Thus, the main challenge of our research is to develop better and more efficient algorithms that are able to give a good and efficient solution for the FBNDP in reasonable computation time.

The procedure starts with a construction algorithm that builds an initial feasible solution from scratch on a stochastic basis. The concept of delimiter is applied when generating the initial solution to prevent too many bad selections such that they are unlikely to form a good solution. This initial solution is then further improved by metaheuristics, which are the most recent development for solving hard combinatorial optimization problems by incorporating concepts derived from artificial intelligence, biological evolution, mathematical and physical sciences to improve their performance. In this research, four metaheuristics: SA, TS, GA and ACO are developed and applied to optimize the FBNDP. All these four metaheuristics are unique in their own ways when searching for the optimal solution. SA is able to avoid getting trapped at a local minimum by probabilistically accepting a neighbourhood solution that is worse than the current one. TS makes use of short and long term memory structures to store certain attributes of recently visited solutions in tabu lists and forbids these attributes for some time to prevent the repetition of moves. The application of SA and TS requires a definition for an appropriate neighbourhood structure that defines the set of solutions that can be reached in one single step from a current solution. We have proposed three new types of neighbourhood moves: (1) The exchange move in which the positions of two stops from two different routes are exchanged. (2) The reduction-displacement move in which a stop is displaced from one route to another existing route. (3) The addition-displacement move in which a stop is removed from one route, allowing it to form a new route by linking it to its nearest station. For TS, two versions are developed: the basic TS which only uses short-term memory and TS with intensification that employs an intensification strategy to exploit the elite solutions discovered in the search process.

In GA, new and better generations of solutions are evolved by going through the selection, crossover and mutation processes. A unique chromosome representation of a solution is proposed such that a chromosome can only be decoded to one solution. A new crossover operator is designed which is able to preserve most of the edges in the parent solutions and inherit the best qualities from the parents, while ensuring the validity of the new offsprings produced. ACO is inspired by the behaviour of real ants which lay pheromone trails on the paths they have travelled to direct the search of the future ants. In our proposed method, each artificial ant constructs a feasible solution on its own using a probabilistic decision rule that makes use of pheromone trails and a priori information about the problem. Due to the multi-station nature of the problem, two probability functions are defined to calculate the desirability of linking each stop to its neighbouring stops and to the stations in the network. A combined elitist and ranking strategy is used and ants deposit pheromone on the edges it has used if it is one of the elitist ants or if the edges travelled by them also belong to the best solution found so far.

Computational results are compared to those published in literature for the base problem. A comparative study is also carried out on 20 test problems generated at random to further compare the performance of the metaheuristics in terms of better computational efficiency and better solution quality. These test problems vary in several characteristics such as the problem size and the problem structure. In general, TS with intensification is the most effective metaheuristic, as it is generating better quality solutions and also obtaining the best solutions for most problems. TS with intensification also produces a new best solution as compared to the literature. However, its computational time is the longest. GA is closely comparable to TS, and they are slightly worse than TS with intensification. SA is fast and offers reasonably good solutions. ACO is able to give satisfactory results. In terms of relative computational times, basic TS is the fastest algorithm. GA and ACO require similar amount of time. In terms of the problem types, the performance of TS with intensification is found to be independent of the various characteristics of the test problems. It is able to outperform other metaheuristics regardless of the nature of the problem, and this shows the robustness of the intensification strategy. ACO is able to surpass some of the results by SA for certain types of problems. This result is encouraging because it is among the first attempts to implement ACO for route design problems and it shows ACO's competitiveness to the state-of-the-art heuristics like SA.

In conclusion, this research has two major contributions to the area of feeder bus route design problem. It is among the first to solve the complex FBNDP using ant colony optimization. In addition, up to the present date, it is also the first to present a comparison of four types of metaheuristics for the FBNDP.

Several directions for future research for solving the FBNDP are possible. Firstly, the design of better metaheuristics can be explored. Our implementation of tabu search still lacks some of the important features of TS such as the use of long-term memory and diversification strategy. For ACO, the encouraging results indicate that there is much potential for further development and improvement. ACO has major advantages over SA, TS and GA. SA and TS rely heavily on the concept of neighbourhood, but defining these neighbourhoods is a very difficult task. ACO does not depend on neighbourhoods because new solutions are generated in every iteration and it does not move from one solution to another. When compared to GA, the difficulty in finding appropriate crossover operators is unnecessary. Thus, future work on ACO approach may further improve its quality for FBNDP, even though the current version cannot compete with TS yet. Secondly, it is also possible to consider modifying the objective function and the constraints of the problem, such as to maximize the profit of the bus operator without sacrificing the service quality. Other extensions include the consideration of the elastic nature of passenger demands and also running computational experiments for larger problem sizes. The use of the 2-opt procedure to optimize each route formed can be improved to the 3-opt procedure. Lastly, the metaheuristics can be applied to other models of the network design problem since they have been proven to be very effective in dealing with such complex problems.

References

Aarts, E. H. L. and J. K. Lenstra (eds). *Local Search in Combinatorial Optimization*. John Wiley & Sons, Chichester. 1997.

Baaj, M. H. and H. S. Mahmassani. TRUST: A LISP Program for the Analysis of Transit Route Configurations, *Transportation Research Record 1283*, Transportation Research Board, Washington, D. C., pp. 125-135. 1990.

Baaj, M. H. and H. S. Mahmassani. An AI-Based Approach for Transit Route System Planning and Design, *Journal of Advanced Transportation*, 25(2), pp. 187-210. 1991.

Baaj, M. H. and H. S. Mahmassani. Hybrid Route Generation Heuristic Algorithm for the Design of Transit Networks. *Transportation Research C*, 3(1), pp. 31-50. 1995.

Bansal, A. N. Optimization of Bus Route Network for a Fixed Spatial Distribution. *In Scientific Management of Transport Systems*, N. S. Jaiswal (ed), North-Holland Publishing Company, Amsterdam, The Netherlands, pp. 346-355. 1981.

Bullnheimer, B., R. F. Hartl and C. Strauss. Applying the Ant System to the Vehicle Routing Problem. In *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization*, ed by S. Voss, S. Martello, I. H. Osman, and C. Roucairol, pp. 285-296. Kluwer Academic Publishers, Dordrecht. 1999a.

Bullnheimer, B., R. F. Hartl and C. Strauss. An Improved Ant System Algorithm for the Vehicle Routing Problem, *Annals of Operations Research*, 89, pp. 319-328. 1999b.

Bullnheimer, B., R. F. Hartl and C. Strauss. A New Rank-Based Version of the Ant System: A Computational Study, *Central European Journal for Operations Research and Economics*, 7(1), pp. 25-38. 1999c.

Byrne, B. F. and V. Vuchic. Public Transportation Line Positions and Headways for Minimum Cost. *In Proceedings of the Fifth International Symposium on the Theory of Traffic Flow and Transportation*, New York, pp. 347-360. 1972.

Byrne, B. F. Public Transportation Line Positions and Headways for Minimum User and System Cost in a Radial Case, *Transportation Research*, 9, pp. 97-102. 1975.

Byrne, B. F. Cost Minimizing Positions, Lengths and Headways for Parallel Public Transit Lines Having Different Speeds, *Transportation Research*, 10, pp. 209-214. 1976.

Ceder A. and N. H. M. Wilson. Bus Network Design. *Transportation Research B*, 20(1), pp. 331-344. 1986.

Ceder, A. and Y. Israeli. User and Operator Perspectives in Transit Network Design, *Transportation Research Record 1623*, Transportation Research Board, Washington, D. C., pp. 3-7. 1998.

Chakroborty, P. and T. Dwivedi. Optimal Route Network Design for Transit Systems Using Genetic Algorithms. *Engineering Optimization*, 34(1), pp. 83-100. 2002.

Chang, S. K. Optimization of Radial Bus Network with Multi-Period and Many-to-Many Demand Patterns, *Journal of Advanced Transportation*, 25(2), pp. 225-246. 1991.

Chang, S. K. and P. M. Schonfeld. Multiple Period Optimization of Bus Transit System, *Transportation Research B*, 25(6), pp. 453-478. 1991.

Chang, S. K. and P. Schonfeld. Welfare Maximization with Financial Constraints for Bus Transit Systems, *Transportation Research Record 1395*, pp. 48-57. 1993.

Chang, S. K. and P. M. Schonfeld. Optimal Dimensions of Bus Service Zones, *Journal of Transportation Engineering, ASCE*, 119(4), pp. 567-585. 1993.

Chien, S. and P. Schonfeld. Optimization of Grid Transit System in a Heterogeneous Urban Environment, *Journal of Transportation Engineering, ASCE*, 123(1), pp. 28-35. 1997.

Chien, S. and P. Schonfeld. Joint Optimization of a Rail Transit Line and Its Feeder Bus System. *Journal of Advanced Transportation*, 32(3), pp. 253-284. 1998.

Chien, S. and Z. Yang. Optimal Feeder Bus Routes on Irregular Street Networks. *Journal of Advanced Transportation*, 34(2), pp. 213-248. 2000.

Chien, S., Z. Yang and E, Hou. Genetic Algorithm Approach for Transit Route Planning and Design. *Journal of Transportation Engineering*, 127(3), pp. 200-207. 2001.

Chien, I. J. and L. N. Spasovic. Optimization of Grid Bus Transit Systems With Elastic Demand, *Journal of Advanced Transportation*, 36(1), pp. 63-91. 2002.

Chowdhury, S. M. and S. I. J. Chien. Intermodal Transit System Coordination. *Transportation Planning and Technology*, 25, pp. 257-287. 2002.

Clarens, G. C. and V. F. Hurdle. An Operating Strategy for a Commuter Bus System, *Transportation Science*, 9, pp. 1-20. 1975.

Davis, L. (ed). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York. 1996.

Dorigo, M., V. Maniezzo and A. Colorni. The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1), pp. 29-41. 1996.

Dorigo, M and T. Stutzle. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances, To appear in *Handbook of Metaheuristics*, by F. Glover and G. Kochenberger. 2001.

Erlander, S. and S. Scheele. A Mathematical Programming Model for Bus Traffic in a Network. *In Proceedings of the Sixth International Symposium on Transportation and Traffic Flow Theory*, pp. 581-605, D. J. Buckley (eds). 1974.

Glover, F. Tabu Search – Part I, *ORSA Journal on Computing*, 1(3), pp. 190-206. 1989.

Glover, F. Tabu Search – Part II, *ORSA Journal on Computing*, 2(1), pp. 4-32. 1990.

Glover, F. and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA. 1997.

Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA. 1989.

Holroyd, E. M. The Optimal Bus Service: A Theoretical Model for a Large Uniform Urban Area. *In Proceedings of the Third International Symposium on the Theory of Traffic Flow*, New York, pp. 309-328. 1967.

Hsu, J. D. and V. H. Surti. Framework of Route Selection in Bus Network Design, *Transportation Research Record 546*, pp. 44-57. 1975.

Hsu, J. D. and V. H. Surti. Decomposition Approach to Bus Network Design, *Journal of Transportation Engineering, ASCE*, 103(4), pp. 447-460. 1977.

Hurdle, V. F. Minimum Cost Locations for Parallel Public Transit Lines, *Transportation Science*, 7(4), pp. 340-350. 1973.

Hurdle, V. F. and S. C. Wirasinghe. Location of Rail Stations for Many to One Travel Demand and Several Feeder Modes, *Journal of Advanced Transportation*, 14(1), pp. 29-45. 1980.

Imam, M. O. Optimal Design of Public Bus Service With Demand Equilibrium, *Journal of Transportation Engineering*, 124(5), pp. 431-436. 1998.

Jansson, J. O. A Simple Bus Line Model for Optimization of Service Frequency and Bus Size, *Journal of Transport Economics and Policy*, 14(1), pp. 53-80. 1980.

Johnson, D. S., J. K. Lenstra and A. G. H. Rinnoykan. *Complexity of the Network Design Problem*. Mathematical Center, Department of Operations Research, Amsterdam. 1977.

Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi. Optimization by Simulated Annealing, *Science*, 220, pp. 671-680. 1983.

Kocur, G. and C. Hendrickson. Design of Local Bus Service with Demand Equilibration, *Transportation Science*, 16(2), pp. 149-170. 1982.

Kuah, G.K. *The Feeder Bus Route Design Problem*. Ph.D Dissertation, University of Maryland, USA. 1986.

Kuah, G. K. and J. Perl. Optimization of Feeder Bus Routes and Bus-Stop Spacing, *Journal of Transportation Engineering, ASCE*, 114(3), pp. 341-354. 1988.

Kuah, G.K. and J. Perl. The Feeder-Bus Network-Design. *The Journal of the Operational Research Society*, Vol. 40, No. 8, pp. 751-767. 1989.

Lampkin, W. and P. D. Saalmans. The Design of Routes, Service Frequencies and Schedules for a Municipal Bus Undertaking: A Case Study, *Operations Research Quarterly*, 18(4), pp. 375-397. 1967.

Lesley, L. J. S. Optimum Bus Stop Spacing - Part 1, *Traffic Engineering and Control*, 17, pp. 399-401. 1976.

Lingaraj, B. P. et al. An Optimization Model for Determining Headways for Transit Routes, *Transportation Planning and Technology*, 3, pp. 81-90. 1976.

Liu, S. Q. and H. L. Ong. A Comparative Study of Algorithms for the Flowshop Scheduling Problem, *Asia-Pacific Journal of Operational Research*, 19, pp. 205-222. 2002.

Mandl, C. E. Evaluation and Optimization of Urban Public Transportation Networks, *European Journal of Operational Research*, 5, pp. 396-404. 1980.

Martins, C. L. and M. V. Pato. Search Strategies for the Feeder Bus Network Design Problem, *European Journal of Operational Research*, 106, pp. 425-440. 1998.

Newell, G. F. Some Issues Relating to the Optimal Design of Bus Routes, *Transportation Science*, 13(1), pp. 20-35. 1979.

Ngamchai, S. and D. J. Lovell. Optimal Time Transfer in Bus Transit Route Network Design Using a Genetic Algorithm. *Journal of Transportation Engineering*, 129(5), pp. 510-521. 2003.

Oldfield, R. H. and P. H. Bly. An Analytic Investigation of Optimal Bus Size, *Transportation Research B*, 22(5), pp. 319-337. 1988.

Osman, I. H. and J. P. Kelly (eds). *Meta-Heuristics: Theory and Applications*. Boston: Kluwer Academic Publishers. 1996.

Pattnaik, S. B., S. Mohan and V. M. Tom. Urban Bus Transit Route Network Design Using Genetic Algorithm. *Journal of Transportation Engineering*, 124(4), pp. 368-375. 1998.

Pham, D. T. and D. Karaboga. *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. London; New York: Springer. 2000.

Rapp, M. H. Transit System Planning: A Man-Computer Interactive Graphic Approach, *Highway Research Record*, 415, pp. 49-61. 1972.

Rapp, M. H., P. Mattenberger, S. Piguet and Robert-Grandpierre. A. Interactive Graphic System for Transit Route Optimization, *Highway Research Record*, 559, pp. 73-88. 1976.

Rea, J. C. Designing Urban Transit Systems: An Approach to the Route Technology Selection Problem, *Highway Research Record*, 417, pp. 49-59. 1972.

Reeves, C. R. (ed). *Modern Heuristic Techniques for Combinatorial Problems*. London; Cambridge, Mass., USA: Blackwell Scientific Publications. 1993.

Sait, S. M. and H. Youssef. *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*. Los Alamitos, CA: IEEE Computer Society. 1999.

Scheele, S. and S. Erlander. A Supply Model for Public Transit Service, *Transportation Research B*, 14, pp. 133-148. 1980.

Shrivastav, P. and S. L. Dhingra. Development of Feeder Routes for Suburban Railway Stations Using Heuristic Approach, *Journal of Transportation Engineering*, 127(4), pp. 334-341. 2001.

Silman, L. A., Z. Barzily and U. Passy. Planning the Route System for Urban Buses, *Computers and Operations Research*, 1, pp. 201-211. 1974.

Spasovic, L. N. and P. Schonfeld. A Method for Optimizing Transit Service Coverage, *Transportation Research Record 1402*, pp. 28-39. 1993.

Spasovic, L. N., M. P. Boile and A. K. Bladikas. Bus Transit Service Coverage for Maximum Profit and Social Welfare, *Transportation Research Record 1451*, pp. 12-22. 1994.

Tan, K. C., L. H. Lee, Q. L. Zhu and K. Ou. Heuristic Methods for Vehicle Routing Problem with Time Windows, *Artificial Intelligence in Engineering*, 15, pp. 281-295. 2001a.

Tan, K. C., L. H. Lee and K. Ou. Artificial Intelligence Heuristics in Solving Vehicle Routing Problems with Time Window Constraints, *Engineering Applications of Artificial Intelligence*, 14, pp. 825-837. 2001b.

Tillman, F. A. and T. M. Cain. An Upper Bound Algorithm for the Single and Multiple Terminal Delivery Problem, *Management Science*, 18, pp. 664-682. 1972.

Tom, V. M. and S. Mohan. Transit Route Network Design Using Frequency Coded Genetic Algorithm. *Journal of Transportation Engineering*, 129(2), pp. 186-195. 2003.

Tsao, S. M. and P. Schonfeld. Optimization of Zonal Transit Service, *Journal of Transportation Engineering, ASCE*, 109(2), pp. 257-272. 1983.

Tsao, S. M. and P. Schonfeld. Branched Transit Service: An Analysis, *Journal of Transportation Engineering, ASCE*, 110(1), pp. 112-128. 1984.

Van Breedam, A. Improvement Heuristics for the Vehicle Routing Problem Based on Simulated Annealing, *European Journal of Operational Research*, 86, pp. 480-490. 1995.

Van Breedam, A. Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem, *Computers & Operations Research*, 28, pp. 289-315. 2001.

Vaughan, R. Optimum Polar Network for an Urban System with a Many-to-Many Travel Demand, *Transportation Research B*, 20(3), pp. 215-224. 1986.

Vaughan, R. J. and E. A. Cousins. Optimum Location of Stops on a Bus Route. *In Proceedings of the Seventh International Symposium on Transportation and Traffic Flow Theory*, pp. 697-716. 1977.

Wilf, H. S. *Algorithms and Complexity*. Englewood Cliffs, N. J.: Prentice-Hall. 1986.

Wirasinghe, S. C. Assignment of Buses in a Coordinated Rail and Bus Transit System. *In Proceedings of the Seventh International Symposium on Transportation and Traffic Flow Theory*, pp. 673-696. 1977.

Wirasinghe, S. C. Nearly Optimal Parameters for a Rail/Feeder Bus System on a Rectangular Grid, *Transportation Research A*, 14(1), pp. 33-40. 1980.

Wirasinghe, S. C., V. F. Hurdle and G. F. Newell. Optimal Parameters for a Coordinated Rail and Bus Transit System, *Transportation Science*, 11(4), pp. 359-374. 1977.

Wirasinghe, S. C. and N. S. A. Ghoneim. Spacing of Bus-Stops for Many-to-Many Travel Demand, *Transportation Science*, 15(3), pp. 210-221. 1981.

Wirasinghe, S. C. and H. H. Ho. Analysis of a Radial Bus System for CBD Commuters Using Auto Access Modes, *Journal of Advanced Transportation*, 16(2), pp. 189-207. 1982.