A Simultaneous Bus Route Design and Frequency Setting Problem for Tin Shui Wai, Hong Kong

W.Y. Szeto [1 *], Yongzhong Wu [2]

Department of Civil Engineering, The University of Hong Kong

Department of Civil Engineering, The National University of Singapore

Abstract

A bus network design problem for Tin Shui Wai, a suburban residential area in Hong Kong, is investigated, which considers the bus services from origins inside this suburban area to the destinations in the urban areas. The problem aims to improve the existing bus services by reducing the number of transfers and the total travel time of the users. This has been achieved by the proposed integrated solution method which can solve the route design and frequency setting problems simultaneously. In the proposed solution method, a genetic algorithm, which tackles the route design problem, is hybridized with a neighborhood search heuristic, which tackles the frequency setting problem. A new solution representation scheme and specific genetic operators are developed so that the genetic algorithm can search all possible route structures, rather than selecting routes from the predefined set. To avoid premature convergence, a diversity control mechanism is incorporated in the solution method based on a new definition of hamming distance. To illustrate the robustness and quality of solutions obtained, computational experiments are performed based on 1000 perturbed demand matrices. The t-test results show that the design obtained by the proposed solution method is robust under demand uncertainty, and the design is better than both the current design and the design obtained by solving the route design problem and the frequency setting problem sequentially. Compared with the current bus network design, the proposed method can generate a design which can simultaneously reduce the number of transfers and total travel time at least by 20.9% and 22.7% respectively. Numerical studies are also performed to illustrate the effectiveness of the diversity control mechanism introduced and the effect of weights for objective functions.

Keywords: Transportation; Bus network design; Route design problem; Frequency setting problem; Genetic algorithm, Neighborhood search

*Corresponding author: Tel.: +852-28578552; Fax: +852-25595337.
E-mail address: ceszeto@hku.hk

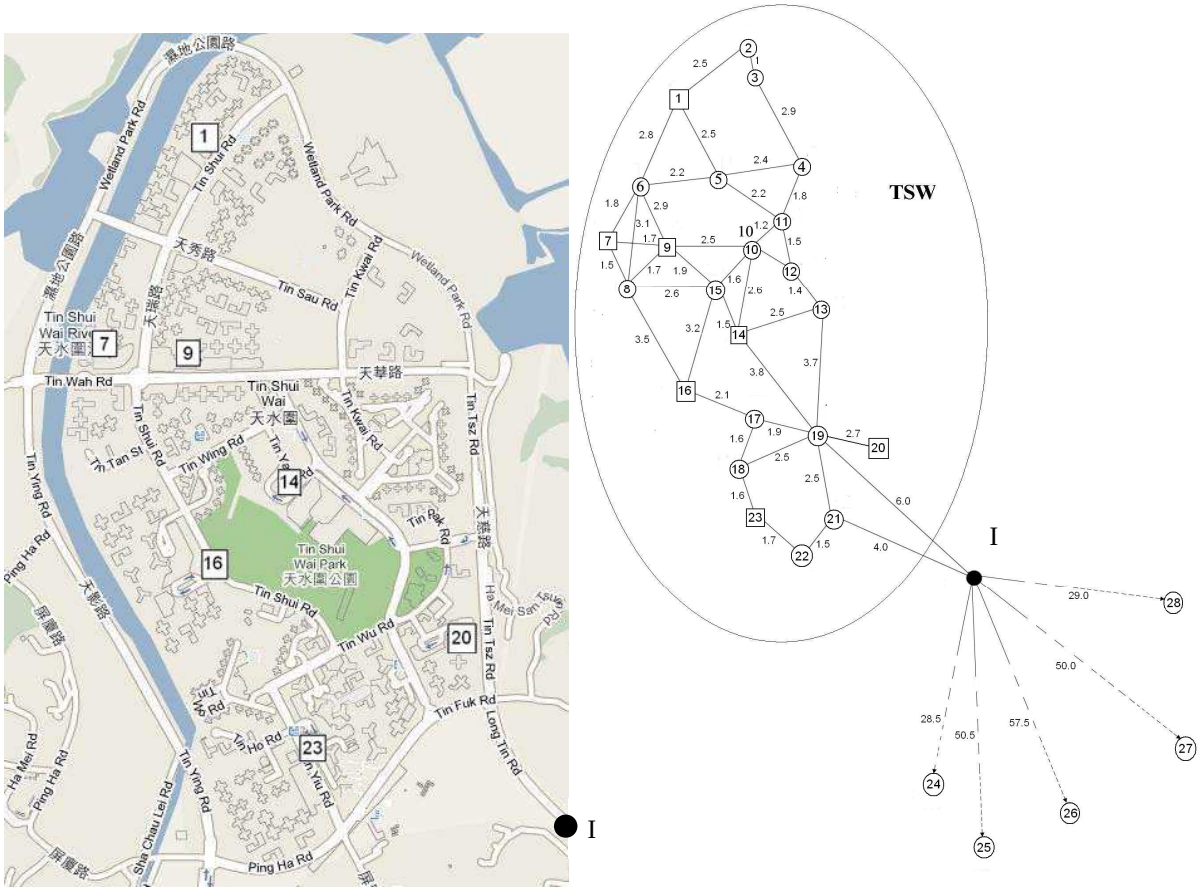
1. Introduction

With continuous population growth of many large cities, like Hong Kong, new suburban residential areas emerge in the periphery of the cities. Transit network design is unique for these suburban residential areas to provide adequate transportation services from the origins inside these areas to the urban working and shopping destinations.

This paper focuses on the trunk bus network design problem for a suburban residential area, Tin Shui Wai (TSW), in Hong Kong shown in Figure 1a. TSW has a population of more than 300,000 people. Most of the residents in TSW work in the urban areas of Hong Kong. Trunk bus is the main commuting mode for TSW residents. The first set of trunk bus routes commenced in the early 1990s. Since then, trunk bus routes in TSW have developed with the growth of the area. Completion of new housing estates led to demand for new routes, which were added on the existing basis. Currently, all the routes go out of TSW to the city centre through the Tai Lam Tunnel (TLT) located at the southeast side of the area. Free transfers at Tai Lam Tunnel bus interchange (indicated as “I”) are allowed.

Being lack of systematic design, the existing bus network operates in an inefficient manner. Transfer at TLT interchange is required by many passengers. Moreover, many bus services are routed to loop around various zones in TSW. This results in an increase in travel time. As a result, the residents complained about the poor performance of the bus service. To react with the complaint, the private operator plans to restructure the bus routes in TSW to reduce both the number of transfers and total passenger travel time. However, as a private bus operator, it highly concerns its profitability, and does not want to increase the operating cost and the fleet size.

The investigated bus network design problem consists of two sub-problems in the sequential transit planning process stated in Ceder and Wilson (1986), namely, route designing and frequency setting. Indeed, many researchers (e.g., Lampkin and Saalmans, 1967; Bel et al., 1979; Carrese and Gori, 2002) adopt the sequential planning approach and tackle the route design problem without considering frequency setting, either because the combined problem is too large to handle or the objectives (e.g., area coverage, route directness, demand satisfaction and total route length) do not depend on frequency. However, as long as the passengers’ total travel time or the number of vehicles is included in the objective function, as is the case for the studied problem, the frequency setting problem should be simultaneously solved with the route design problem because both the passengers’ total travel time and the number of vehicles depend on frequency.



(a) Map of Tin Shui Wai (b) The Tin Shui Wai bus network

Figure 1 The investigated network

Only a few researchers have simultaneously tackled both the route design and frequency setting problems, which is classified as the transit network design and frequency setting problem according to a recent, comprehensive, and systematic transit network design review of Guihaire and Hao (2008). Instead of solving the two problems in a sequential manner, Lee and Vuchic (2005) adopt the iterative approach to handle the interaction between the two sub-problems. In a similar way, Pacheco et al. (2009) solve the route design problem and the bus assignment problem in alternating steps, while each solution is tackled by a separate procedure. Although good results can be obtained in these works, the convergence of this approach is questionable. Some attempts have been made to use exact methods to solve the combined problem (e.g., Bussieck, 1998; Wan and Lo, et al., 2003; Borndöfer et al., 2005). However, these methods cannot solve large realistic problems due to the NP-hard nature of these problems. Therefore, to solve these problems, most researchers (e.g., Van Nes et al., 1988; Shih and Mahmassani, 1994; Shih et al. 1998; Pattnaik et al., 1998; Bielli et al. 2002; Fusco et al., 2002; Tom and Mohan, 2003; Ngamchai and Lovell, 2003; Fan and Machemehl,

2006a, 2006b; Zhao and Zeng, 2007) employ heuristics or meta-heuristics, which require generate an initial candidate route set. Nevertheless, the initial route set may affect the quality of the final solution (which can highly depend on the initial solution for some solution methods) and the suitability may vary from network to network. Moreover, whether to include a single stop on a route can greatly affect both the number of transfers and travel time. Therefore, it is difficult to define an initial set of candidate routes for these methods.

The bus network design problem investigated in this paper simultaneously considers both route design and frequency setting, and is characterized by two aspects that make our problem different from other combined transit route design and frequency setting problems in the literature. First, the objectives and the constraints of the problem are different from those in the existing literature mentioned before. Second, the layout and the design of studied route network are distinct. The investigated bus network design involves planning trunk bus routes to connect the suburban area to the urban area. All the routes must pass through one large interchange but there is no restriction on the number of bus lines serving any demand point inside the suburban area. To our best knowledge, this feature cannot be found in other transit network design including bus feeder system design (e.g., Martins and Pato, 1998 and Kuan et al., 2006) in which the interchanges are next to rail stations, and each bus stop in the suburban area is only served by one feeder bus service.

This paper also proposes a solution method that considers all possible route structures with all possible stop combinations. More importantly, this solution method developed here is different from the GAs and hybrid GAs proposed in the literature. A diversity control mechanism based on a new definition of hamming distance is proposed to solve for solutions.

The remaining parts of this paper are organized as follows: Section 2 presents the formulation of the problem. Section 3 describes the proposed solution method. Section 4 depicts computational results. Finally, section 5 gives concluding remarks and highlights future research directions.

2. Formulation of the problem

The TSW network is shown in Figure 1b. The square nodes represent the locations with both bus terminals and bus stops; the circle nodes represent the current bus stop locations, and "L" represents the TLT bus interchange. The in-vehicle travel times (in minutes) between nodes are shown next to the corresponding links. As shown in Figure 1b, the TSW area has 23 stops (i.e., nodes 1 - 23 in the figure) and there are seven bus terminals in this area. All the bus routes originated from these terminals terminate at one of the five destinations, nodes 24 - 28.

The demand matrix estimated from the available data is given in Appendix 1.

Ideally, there should be as many routes as possible so as to provide direct point-to-point services for the residents. However, this is not feasible due to relatively fixed operating cost of the operator (like other bus operators, the operating cost is shown to be roughly proportional to the number of operating vehicles). In addition, if a single route is too zigzag and has too many stops, the travel time is long. What is intended in this work is a complete restructuring of the bus network inside this area to reduce the total number of transfers and the total travel time (including waiting time) without increasing the number of operating vehicles, and without allowing too long passengers' travel times and too many stops.

To formulate and generalize this real but specific problem for other similar applications, we introduce more notations than necessary in this paper, and expand the TSW network to include 1 dummy node (i.e., node 0) and 14 dummy links. The dummy links are used to connect the dummy node and bus terminals/destinations. The notations and the parameters of this study are given below:

Sets/Indices

Z: set of nodes excluding the dummy node;

U: set of demand locations (i.e., stops and bus terminals inside TSW);

V: set of the destinations outside TSW, i.e., nodes 24 – 28;

Y: set of the bus terminals inside TSW, i.e., nodes 1, 7, 9, 14, 16, 20, and 23;

C: set of interchange, i.e., node L;

i, j, k, e: indices of nodes.

Parameters

c_{ij} = in-vehicle travel time on the shortest path between nodes i and j;

s = average time for stopping at a node, i.e., 1.5 minutes;

d_{ie} = travel demand from node i to destination e;

W = maximum bus fleet size, i.e., 176;

R_{max} = maximum number of routes on the bus network, i.e., 10;

f_{min} = minimum frequency of a route, 4.8 buses per hour;

S_{max} = maximum number of intermediate stops on the TSW portion of a route, i.e., 8;

T_{max} = maximum travel time from the bus terminal in TSW to TLT interchange, including stopping time), i.e., 35 minutes;

B_1 = weight for the number of transfers;

B_2 = weight for the total travel time.

Decision Variables

X_{ijn} = 1 if route n (n = 1 to R_{max}) passes through node $j \neq i$ immediately after node i; = 0 otherwise;

X_{0jn} = 1 if route n starts at node j; = 0 otherwise;

X_{i0n} = 1 if route n ends at node i; = 0 otherwise;

X_{00n} = 1 if route n is not available; = 0 otherwise;

RT_{ij}^{n} = l if route n passes through node i and node j; = 0 otherwise;

NR_{ie} = l if there is no direct bus service from node i to destination e; = 0 otherwise;

f_n = frequency of route n;

T_n = single trip time of route n;

T_{ie} = average travel time of passengers from stop i to destination e;

T_{ij}^{n} = travel time from node i to node j via route n.

Based on the above notations, the mathematical model is formulated as follows:

$$\min z = B_1 \sum_{i \in U} \sum_{e \in V} d_{ie} NR_{ie} + B_2 \sum_{i \in U} \sum_{e \in V} d_{ie} T_{ie} \tag{1}$$

subject to

$$\sum_{j \in Y \cup \{0\}} X_{0jn} = 1 \qquad \text{for } n = 1 \text{ to } R_{max}, \tag{2}$$

$$\sum_{i \in V \cup \{0\}} X_{i0n} = 1 \qquad \text{for } n = 1 \text{ to } R_{max}, \tag{3}$$

$$\sum_{i \in Z \cup \{0\}, i \neq j} X_{ijn} - \sum_{i \in Z \cup \{0\}, i \neq j} X_{jin} = 0 \qquad \text{for } j \in Z, \quad n = 1 \text{ to } R_{max}, \tag{4}$$

$$\sum_{i \in Z \cup \{0\}} X_{ijn} \leq 1 \qquad \text{for } j \in Z, \quad n = 1 \text{ to } R_{max}, \tag{5}$$

$$\sum_{i \in Z \cup \{0\}} X_{ijn} \leq 1 \qquad \text{for } i \in Z, \quad n = 1 \text{ to } R_{max}, \tag{6}$$

$$T_n = \sum_{i \in Z} \sum_{j \in Z, j \neq i} X_{ijn}(c_{ij} + s) - s \qquad \text{for } n = 1 \text{ to } R_{max}, \tag{7}$$

$$\sum_{n=1}^{R_{max}} 2 f_n T_n (1 - X_{00n}) \leq W, \tag{8}$$

$$f_{min}(1 - X_{00n}) \leq f_n \qquad \text{for } n = 1 \text{ to } R_{max}, \tag{9}$$

$$\sum_{i\in U}\sum_{j\in U,\, j\neq i} X_{ijn} \le S_{\max} \qquad \text{for } n = 1 \text{ to } R_{\max}, \tag{10}$$

$$T_{ie} = \frac{\sum_n RT_{ie}^n f_n(T_{ie}^n)+1}{NR_{ie}+\sum_n RT_{ie}^n f_n}\left(1-NR_{ie}\right) + \left(\frac{\sum_n RT_{ik}^n f_n T_{ik}^n+1}{\sum_n RT_{ik}^n f_n} + \frac{\sum_n RT_{ke}^n f_n T_{ke}^n+1}{\sum_n RT_{ke}^n f_n}\right) NR_{ie}$$

$$\text{for } i\in U,\ k\in C,\ e\in V, \tag{11}$$

$$T_{ik}^n = X_{ikn}c_{ik} + \sum_{j\in Z/V,\, j\neq i,\, j\neq k} X_{ijn}RT_{jk}^n\left(T_{jk}^n + c_{ij} + s\right)$$

$$\text{for } i\in Z/V,\ k\in Z/U,\ k\neq i,\ n = 1 \text{ to } R_{\max}, \tag{12}$$

$$RT_{ik}^n = X_{ikn} + \sum_{j\in Z/V,\, j\neq i,\, j\neq k} X_{ijn}RT_{jk}^n \qquad \text{for } i\in Z/V,\ k\in Z/U,\ k\neq i,\ n = 1 \text{ to } R_{\max}, \tag{13}$$

$$NR_{ie} = \prod_{n=1}^{R_{\max}}\left(1-RT_{ie}^n\right) \qquad \text{for } i\in U,\ e\in V, \tag{14}$$

$$\sum_{i\in U\cup C}\sum_{j\in U\cup C,\, j\neq i} X_{ijn}(c_{ij}+s) - s \le T_{max}^{'} \qquad \text{for } n = 1 \text{ to } R_{max}. \tag{15}$$

The objective of the problem is to minimize the weighted sum of the number of transfers and total passengers' travel time (including in-vehicle travel time and waiting time). This traditional weighted sum approach is used to formulate the objective because of the two advantages of this approach: First, the resultant model can give one answer per run directly once the decision makers can clearly define the relative importance of the two objectives; there is no need to do another analysis to determine the best solution from a set of Pareto solutions as in the bi-objective problem. Second, the computation time for the bi-objective problem is higher in general. Therefore, we adopt the weighted sum approach and leave the bi-objective approach for future studies.

Constraint (2) ensures that each available route starts from one of the TSW terminals (i.e., the squares shown as in Figure 1). Constraint (3) ensures that each available route ends at one of the bus terminals/destinations outside TSW. Constraint (4) ensures that except dummy nodes, any node on an available route must have one preceding node and one following node. Constraints (5) and (6) ensure that each node can only be visited by a particular available route at most once. Constraint (7) calculates the in-vehicle travel time (including stop time) of a route. Constraint (8) ensures that the fleet size cannot exceed the maximum fleet size. Constraint (9) depicts the minimum frequency requirement. Constraint (10) limits the number of the intermediate stops.

The average travel time from node i to destination e, T_{ie}, is a complex function of the

route structure and frequencies, which is defined by Constraint (11). The calculation of T_{ie} is based on the assumption of the demand assignment in Baaj and Mahmassani (1990): 1) the passengers go to their destinations using minimal transfers; 2) the passengers get into the first arrived bus among a set of attractive lines, and; 3) the demand is split according to the frequencies of attractive lines. Based on these assumptions, the T_{ie}, value is determined in the following way: If there are direct bus services serving between stop *i* and destination *e*, the expected travel time T_{ie} is the weighted average in-vehicle travel time of all these direct services plus the expected waiting time for a bus operating for these services; if there is no direct service serving between stop *i* and destination *e*, the T_{ie} value is the sum of expected travel time from stop *i* to TLT interchange and that from TLT interchange to destination *e*. The travel time from node *i* to node *j* via route *n*, T_{ij}^{n}, is calculated by Constraint (12). Constraint (13) defines RT_{ij}^{n}, which is l if route *n* passes through both nodes *i* and *j*, and 0 otherwise. Constraint (14) defines NR_{ie}, which is l if there is no direct service serving between node *i* and destination *e*, and 0 otherwise. Constraint (15) ensures that the travel time from each bus terminal to TLT interchange is not greater than the maximum allowable travel time.

It can be seen that constraints (8), (11) and (15) are all nonlinear. Moreover, the decision variables contain both integer and continuous variables. Therefore, the formulation is a mixed-integer nonlinear program. It has been shown that a general transit network problem without considering frequency is already *NP*-hard (Magnanti et al., 1984). In order to solve the investigated problem, a heuristic or meta-heuristic is needed.

3. The solution method

3.1. General scheme of the hybrid genetic algorithm

In the proposed solution method, a specific genetic algorithm (GA) is developed to solve the route design problem, while a frequency setting heuristic based on neighborhood search is integrated into the GA to solve the frequency setting problem.

Figure 2 illustrates the scheme of the hybrid GA. First, initial solutions representing route structure are randomly generated. Then, the frequency setting heuristic is performed to evaluate the fitness of the GA solutions, since the GA solutions themselves are not enough for calculating the objective value that is also a function of bus frequencies. After that, some parents are selected using the roulette wheel selection method. A certain number of offspring are generated through genetic operators, i.e., crossover and mutation operators. For each pair of parents, a crossover operator is applied to generate a pair of children. Therefore, the

number of offspring generated is the same as that of the parents. Then, all the offspring will be mutated in order to maintain the diversity of the population. After the mutation, a stop frequency heuristic is applied to the offspring to improve the sequence of stops for each route. Then, a repair operator is applied to those offspring that violate the constraint of the maximum number of intermediate stops and the constraint of maximum allowable travel time within TSW district. The individuals who can survive into the next generation are selected from both the parents and the offspring through a diversity control mechanism. The process is repeated until a preset number of generations is produced.

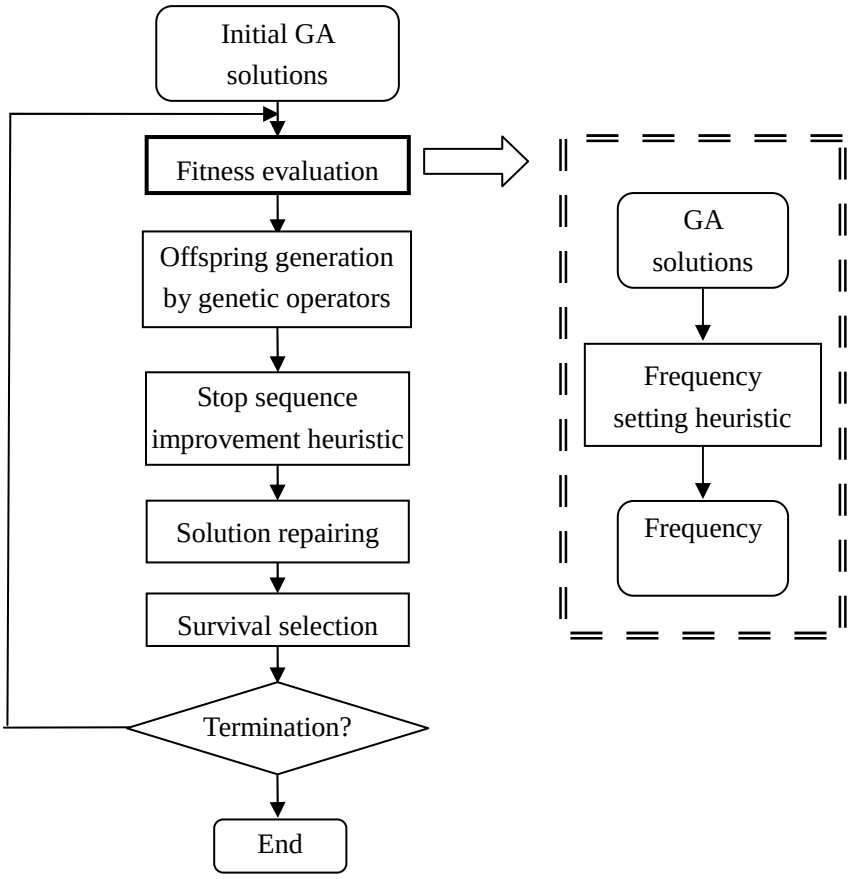


Figure 2 Illustration of the detailed scheme of the genetic algorithm

3.2 Representation scheme and initialization method

A GA solution represents a set of routes, each with a sequence of stops. Figure 3 illustrates the solution representation scheme of the GA. Circles represent stops whereas arrows show the directions of movements. The chromosome consists of 10 routes. The first route starts from node 1, goes through nodes 18, 15, 8, 12, 7, and TLT interchange (which is implicitly coded), and terminates at destination 25. Similarly, the tenth route starts from node 7, goes through nodes 12, 5, 13, 3, and TLT, and finally stops at destination 27.

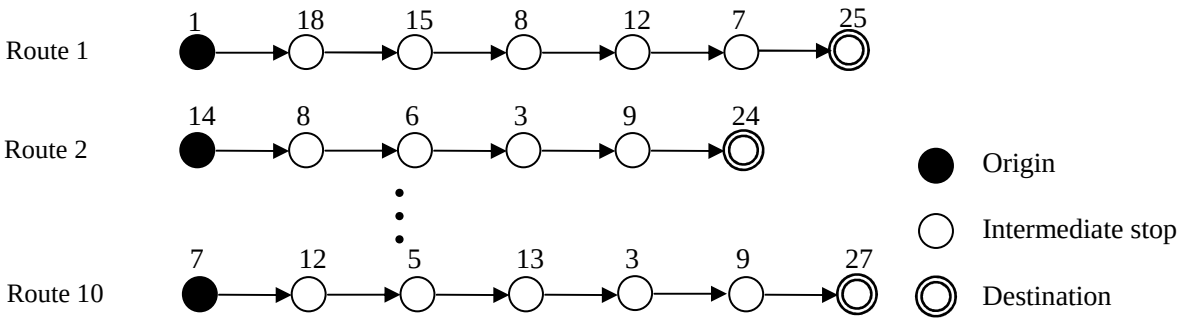


Figure 3 Representation scheme of the genetic algorithm

To initialize a feasible solution, the following procedure is adopted.

For route n = 1 to R_{max}

Randomly select a terminal node and a destination node for route n

Set *time_exceed = false*

Repeat

Randomly select a node i that is not appeared on route n

Insert node i to a position between the positions for the origin and destination in the chromosome such that the travel time of route n is minimized.

If the travel time on the TSW portion of route n exceeds the maximum allowable value, then remove node i from the route and set *time_exceed = true*

Until maximum number of intermediate stops is reached or *time_exceed = true*

Next route

3.3 Frequency setting heuristic and fitness evaluation

The frequency setting heuristic is adopted whenever the fitness of a GA solution is evaluated. This heuristic is designed to solve the frequency setting problem. Because the main constraint in the frequency setting problem is the fleet size constraint, we can reformulate the frequency setting problem into the problem of allocating the limited number of buses to the routes defined by a GA solution. Therefore, instead of explicitly coding the frequencies of these routes, the solution structure for the frequency setting heuristic is defined by a collection of the numbers of buses allocated to each route. Figure 4 illustrates a solution of the frequency setting heuristic. The solution represents a fleet size of 176 buses allocated to 10 routes. For example, the third gene with a value of 16 means that there are 16 buses allocated to the third route. As the total number of buses used in the network is not changed, the fleet size constraint can be ensured. The simplicity of this representation helps avoid creating infeasible

solutions and thus facilitates the searching process.

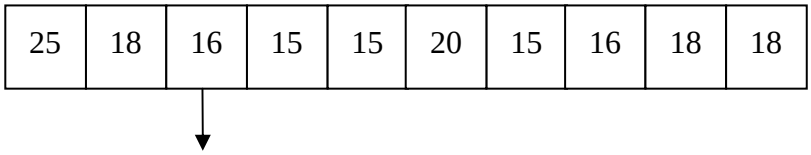


Figure 4 A solution of the frequency setting heuristic

Based on the solution structure shown in Figure 4, a decent search heuristic is developed to obtain a set of frequencies for a given GA solution. This search heuristic is described as follows:

Step 1: Initialize a solution by randomly allocating buses to R_{max} routes

Step 2: For i =1 to R_{max}

For $j = i$ to R_{max}

Move 1 bus from route i to route j

Evaluate the new solution

If the value of the objective function (1) is improved, then go to Step 2. Otherwise, undo the movement

Move 1 bus from route j to route i

Evaluate the new solution

If the value of the objective function (1) is improved, then go to Step 2. Otherwise, undo the movement

Next j

Next i

Step 3: Output the best solution found

The determination of the objective value in the above heuristic relies on the frequency information derived from the solution of the heuristic and the GA solution through:

$$f_n = \frac{V_n}{2T_n} \tag{15}$$

where T_n is the trip time of route n deduced by (7); f_n and V_n are the frequency of route n and the number of buses allocated to route n.

After the heuristic is employed, the fitness of a GA solution is obtained by finding the reciprocal of the objective value. To ensure that the minimum frequency requirement is met, a

special treatment is adopted while calculating the objective value. If a frequency of a route is smaller than the minimum frequency, a penalty is introduced through dividing the frequency by a nonnegative parameter to increase the objective value. In this way, the GA algorithm may gradually throw away all the infeasible solutions in a gentle manner, and the efficiency of the algorithm can be maintained.

3.4 Crossover and mutation operators

Due to the complexity of the problem, specific crossover and mutation operators are developed to improve the effectiveness of the reproduction process. Since the problem consists of two levels, i.e., to determine the optimal combination of routes and to determine the optimal combination of sequences of stops, different crossover operators should be designed to facilitate the exchange of good building blocks of the solutions. For this purpose, two crossover operators are proposed. They are the route and stop crossover operators.

The route crossover operator is to exchange routes between two solutions. It is similar to the traditional two-point crossover. Rather than exchanging individual genes, the route crossover operator exchanges the routes between the two parents, where routes are considered as building blocks. The purpose of this operator is to explore different combinations of routes. In the example shown in Figure 5, routes 5, 6, 7, and 8 of Parent 1 and Parent 2 are exchanged to produce Child 1 and Child 2. If the simple crossover or two-point crossover were used instead, a route forming a good building block could be destroyed easily as the crossover points were not likely located at the positions of genes containing starting bus terminals.

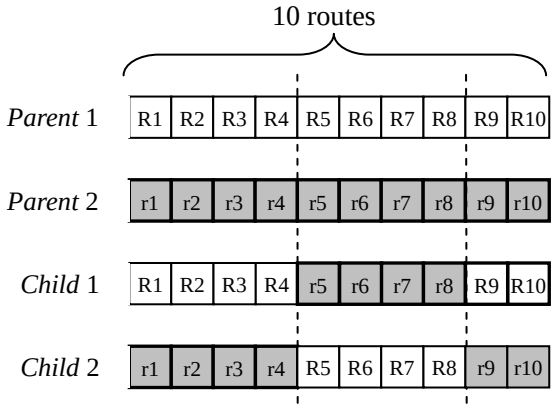


Figure 5 Illustration of route crossover

Comparatively, the stop crossover operator is to exchange sequences of intermediate stops between two routes with the same destination in the two parents, where the sequences of intermediate stops are viewed as smaller building blocks. The purpose of this operator is to

explore different route structures and stop sequences with the same destination to lower total travel time but maintain the same number of transfers. If the same destination requirements were not imposed, good building sub-blocks with respect to the number of transfers could be destroyed easily.

The stop crossover operator works as follows: First, one route is randomly selected from one of the parents. Then, another route with the same destination as the first route is randomly selected from the other parent. After that, a sequence of intermediate stops with arbitrary length in each of the two routes is randomly determined and exchanged. Finally, duplicated stops in the same route are eliminated. Figure 6 illustrates of the process of the stop crossover operator. Route *i* in parent 1 and route *j* in parent 2 having the same destination (i.e., node 25) are selected. Then, sequence 8-12 in route *i* and sequence 6-3-9 in route *j* are selected. The two sequences of stops are exchanged. Finally, because Stop 8 appears twice on route *j* of Child 2, the original one is eliminated from the route.

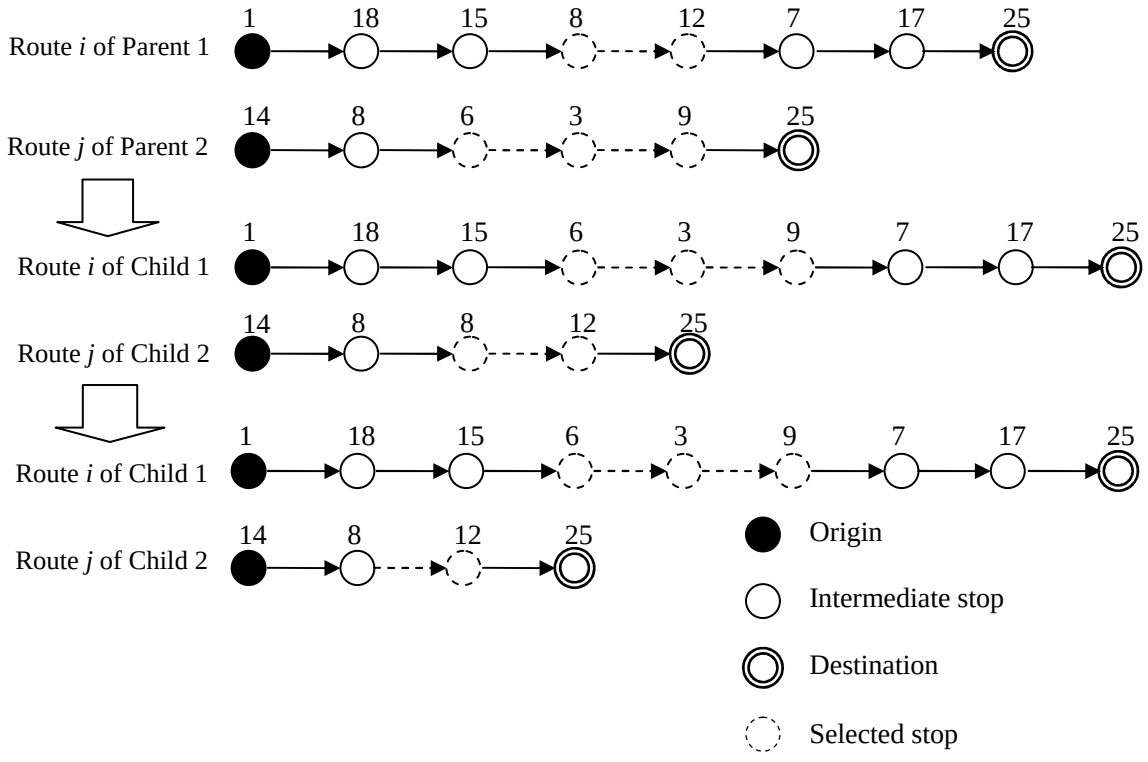


Figure 6 Illustration of stop crossover

Due to the complex solution structure of the problem, different mutation operators are required to facilitate exploration in the large search space. The four mutation operators, namely, the insert, delete, swap, and transfer mutations, are proposed for the GA, which are illustrated in Figure 7. For the insert mutation operator, a stop node is inserted into a route of

the solution. The stop, the route, and the inserted position are all randomly determined. Likewise, the remove mutation operator randomly removes a stop from a route in the solution. Both the insert mutation and remove mutation are intended to create perturbation to a single route. By contrast, the swap and the transfer mutation operators are intended to cause exchange nodes between two routes in a solution. The swap mutation operator exchanges two nodes between two routes of the solution. The exchanged nodes can be starting terminals, intermediate stops, or destinations, but must be of the same type. The transfer mutation operator randomly moves an intermediate stop on one route to another in the solution. For all the mutation operators, a checking mechanism is adopted to avoid the same node appearing on the same route more than once.

3.5 Stop sequence improvement heuristic

To improve the solution quality of the GA, a stop sequence heuristic is proposed for improving the sequence of intermediate stops of each route. For every GA solution that survives into the next generation, a decent search heuristic is adopted to improve the sequence of stops on each route. To ensure the efficiency, the objective of this sequence improving process is to minimize the trip time of the route that does not depend on frequencies and thus is relatively easy to obtain. The process of the heuristic is as follows:

For each route in the GA solution
1 For *i* = the first intermediate stop to the second last intermediate stop
For *j* = *i* to the last intermediate stop
Exchange *i* and *j*
Evaluate the trip time of the route
If the trip time is reduced, then go to 1; otherwise, undo the exchange
Next *j*
Next *i*
Next route in the GA solution

3.6 Repair operator

After genetic operators and stop sequence improvement heuristic are applied to offspring, some offspring may violate the constraint of the maximum number of intermediate stops or the constraint of the maximum allowable travel time. Therefore, a repair operator is implemented for repairing these infeasible solutions.

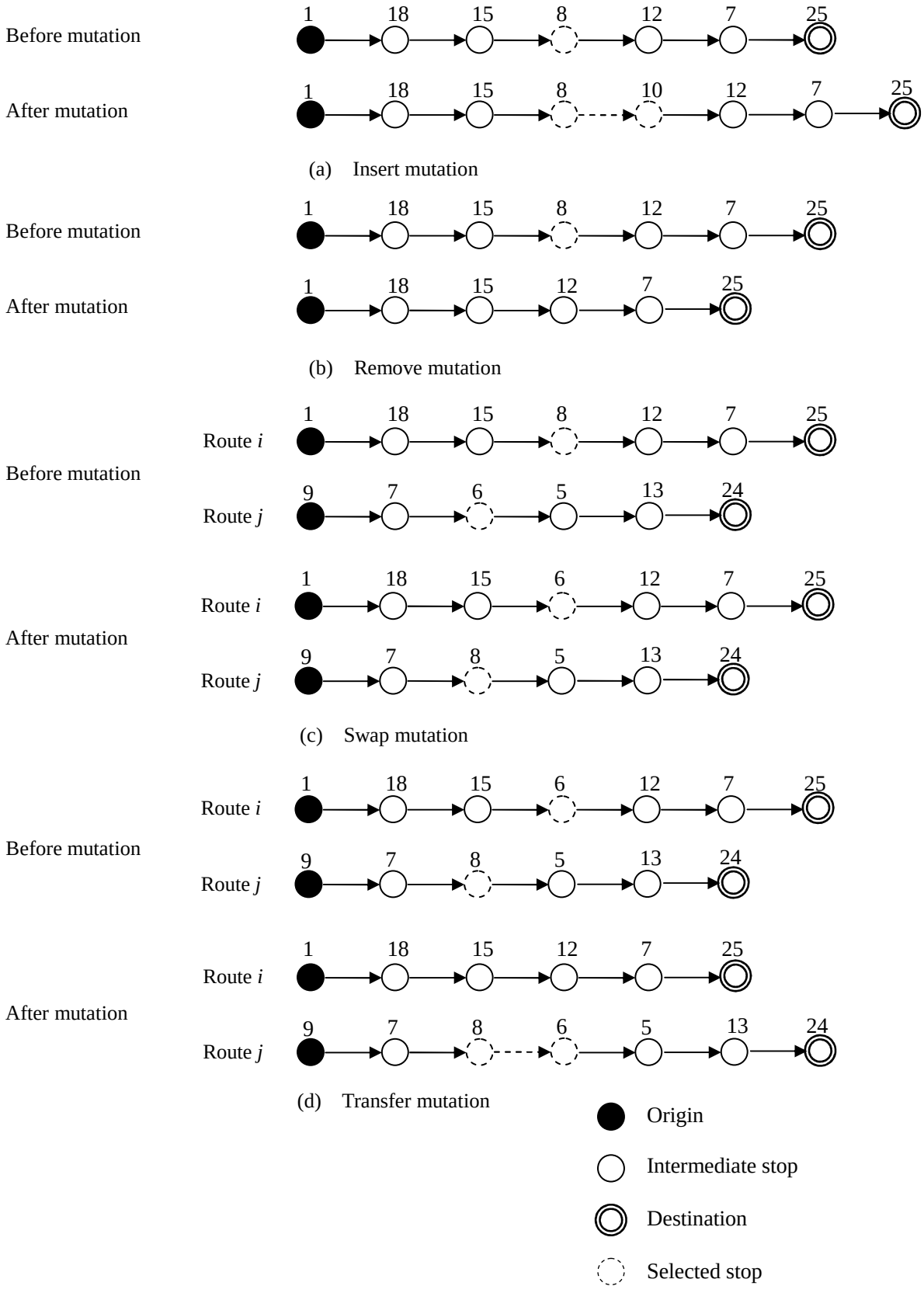


Figure 7 Mutation operators for the genetic algorithm

The repair operator works in the following way. For each route that violates at least one of

the two constraints, determine the intermediate stop, whose removal from the route causes maximum reduction in the trip time of the route. Then, remove this intermediate stop from the route. This stop removal process is continued until the concerned constraint is satisfied.

3.7 Diversity control mechanism

Because the search space is large, how to maintain population diversity to avoid premature convergence is important for the performance of the GA. Therefore, other than using mutation operators, we employ a diversity control mechanism which is similar to that proposed by Shimodaira (2001). In the DCGA proposed by Shimodaira (1997, 2001), the probability of selecting an individual for survival, P_s, is based on its hamming distance from the best individual:

$$P_s = \left\{ (1-c)\frac{h}{L} + c \right\}^{\alpha} \tag{16}$$

where h is the hamming distance between the individual and the best individual. L is the length of the entire string representing the individual. c is the shape coefficient whose value is in the range of [0.0, 1.0], and α is the exponent. According to the given problem, an appropriate selection pressure can be obtained through adjusting c and a in equation (16). For a complicated problem with many local optima, a lower selective pressure can be produced with a smaller value of c and /or a larger value of a. The values of c and a must be tuned by trial and error according to a given problem (Shimodaira, 2001).

The hamming distance is originally defined for binary representation. An extended hamming distance has been defined for permutation representation for problems like the traveling salesman problem (Shimodaira, 1999). However, since the representation in the proposed GA is unique and complicated (which considers multiple routes with different lengths), we propose a new hamming distance for our GA, which is calculated as follows:

1. For each pair of corresponding routes in the two individuals, calculate the distance between two routes, which is defined as the number of different consecutive node pairs in the two routes.
2. The hamming distance of the two individuals is calculated as the sum of all the distances between the two corresponding routes in the two individuals.

For example, Figure 8 illustrates route *i* of individual 1 and route *i* in individual 2. There are totally 5 different consecutive node pairs in these two routes, which are: 18→15, 15→8, 18→6, 6→3, and 3→8. Therefore, the distance between these two routes is 5. The hamming

distance between the two individuals is the sum of the distances between all route pairs (i.e., 10 route pairs for the investigated problem) in the two individuals.

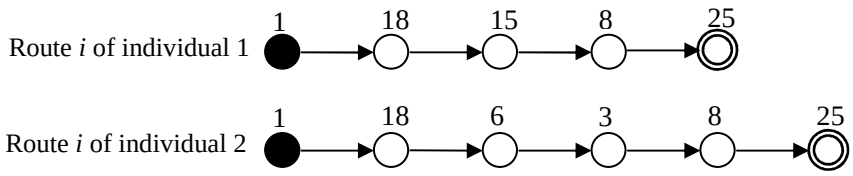


Figure 8 Illustration of the calculation of the distance between two routes

Based on our specifically designed hamming distance, the survival selection process with diversity control proposed by Shimodaira (2001) can be modified and used. The modified survival selection is described as follows:

(1) The children and parents are merged and sorted in order of their fitness values to form the population *M*.

(2) Duplicate individuals in *M* are eliminated.

(3) Select the best individual in *M* for survival into the next generation

(4) Select individuals in *M* (except the best individual) in the order of their fitness values based on the probability calculated by equation (16), where *h* is the hamming distance between the concerned individual and the best individual in *M*, and *L* is re-defined as the total number of consecutive node pairs in these two individuals.

(5) If the number of the individuals selected in step (4) is smaller than the population size, then new individuals randomly generated as in the initialization process are introduced by the difference of the numbers.

4. Experiments

Using the information given in Section 3, computational experiments were conducted to 1) tune GA parameters, 2) illustrate the effect of weights on the objective values, 3) demonstrate of the effectiveness of the proposed diversity control on solution quality, and 4) illustrate the quality of the solution obtained. The proposed solution method was coded in Visual C++ 2003, and ran on a computer with 1.73 GHz CPU and 1 G RAM. In all experiments, the in-vehicle travel times between each pair of nodes was obtained by a shortest path algorithm.

4.1 Fine tuning of the GA

Experiments were conducted to determine the suitable combination of the operators

discussed in the previous section. Totally, eight sets of GAs with different settings were implemented. Each GA was run 20 times. For all these eight GAs, the population size is set to 20. In each generation, 16 offspring are generated through genetic operators. Then, 20 individuals are randomly selected from the 20 parents and 16 offspring to form the population in the next generation. The best 20 out of 36 individuals can survive into the next generation. The weights for the number of transfers and the total travel time, i.e., B_1 and B_2, are set to be 80 and 1, respectively. This combination is used because both total travel time and the number of transfers are lowered by using this combination based on the tuning study in section 4.2.

Each GA terminated when 500 generations were produced. It should be noted that from our preliminary study (not shown here due to space limitation), the stop sequence improvement heuristic proposed in Section 3.4 can improve the objective values by 5-8%. Therefore, the stop sequence improvement heuristic was included in all the experiments.

Table 1 summarizes the computational results of these experiments. For all the eight GAs (GA-1 to GA-8), the computational time is similar and acceptable, and is about 3 minutes per run. GA-1 only uses route crossover to generate offspring, while GA-2 only uses stop crossover. For these two GAs, no mutation operator is used. It can be seen that GA-1 and GA-2 have similar performance in terms of the fitness value. GA-3 uses both crossover operators (without using any mutation operators). i.e., for each pair of selected parents, either route crossover or stop crossover is randomly selected with the same probability. It can be seen that GA-3 achieves a better fitness value than the previous two GAs, indicating that it is advantageous for incorporating the two crossover operators in the algorithm. GA-4, GA-5, GA-6, and GA-7 are the same as GA-3 except that after using crossover operators, they also adopt the insert, delete, swap, and transfer mutation operators respectively. All the four GAs have a better performance than GA-3. For GA-4 which uses insert mutation, the number of transfers is the smallest, because insert mutation increases the number of intermediate stops on the selected routes without decreasing the number of intermediate stops of other routes. However, the lower number of transfers is obtained with the sacrifice of the total travel time. Comparatively, GA-5 which uses delete mutation has the reverse effect. GA-7 which uses transfer mutation seems to perform the best among the four GAs as it has the highest average fitness value and the smallest standard deviation of fitness values. In GA-8, the four mutation operators and two crossover operators are incorporated in the algorithm: For each child randomly generated by either route or stop crossover operator with equal probability, one of the four mutation operators is randomly applied on the child based on the selection probabilities of the mutation operators. The selection probabilities of the insert, delete, swap

and transfer mutation operators are 0.4, 0.4, 0.1, and 0.1, respectively. This combination of selection probabilities is the best for this study and was fine-tuned in a preliminary study. For other studies, it is recommended to tune the selection probabilities to obtain the best result.

Table 1 Computational results of different GA settings

	GA-1	GA-2	GA-3	GA-4	GA-5	GA-6	GA-7	GA-8
Crossover operator used	Route	Stop	Route + stop	Route + stop	Route + stop	Route + stop	Route + stop	Route + stop
Mutation operator used	No	No	No	Insert	Delete	Swap	Transfer	All four
fitness(10^{-7})[a]	4.64	4.67	4.84	5.09	4.87	5.11	5.17	5.27
std. dev.[b]	0.11	0.20	0.11	0.09	0.11	0.12	0.06	0.09
transfers[c]	11308	11083	9979	7652	10739	9174	8855	7700
std. dev.[b]	919	1703	874	699	893	868	398	633
travel time[d]	1253225	1259593	1268335	1353537	1197380	1225917	1227319	1282480
std. dev.[b]	24988	45988	25352	26597	26934	25745	11075	18570
CPU time[e]	3.21	3.28	3.19	3.23	3.31	3.15	3.04	3.24
std. dev.[b]	0.07	0.06	0.05	0.05	0.21	0.05	0.06	0.44

[a] Average fitness value for 20 runs

[b] Standard deviation

[c] Average number of transfers for 20 runs

[d] Average total travel time (in minutes) for 20 runs

[e] Average computational time (in minutes) of each run

From the results in Table 1, it can be seen that GA-8 obtained the best fitness value among all the GAs. It indicates that it is advantageous for incorporating all mutation operators in the algorithm. Indeed, other combinations of selection probabilities have been tested and the conclusion is the same. It is beneficial to use all mutation operators in the algorithm. However, GA-8 gives the best performance. Therefore, in the remaining experiments, the setting of GA-8 will be used.

Figure 9 illustrates the convergence process of the algorithm over 3000 generations. It can be seen that, although the population size is only 20, the algorithm requires about 2000 generations for convergence, indicating the great complexity of the algorithm. A replicate of the GA with a population size of 50 was also run for 3000 generations. The solution found is much similar to those obtained by GA-8 shown in Table 1 but the computation time is increased greatly from 3.24 minutes to 31.7 minutes (or by more than 10 times). This indicates that increasing the population size may greatly extend the convergence process. For this reason, in the following experiments, the population size is kept to 20.

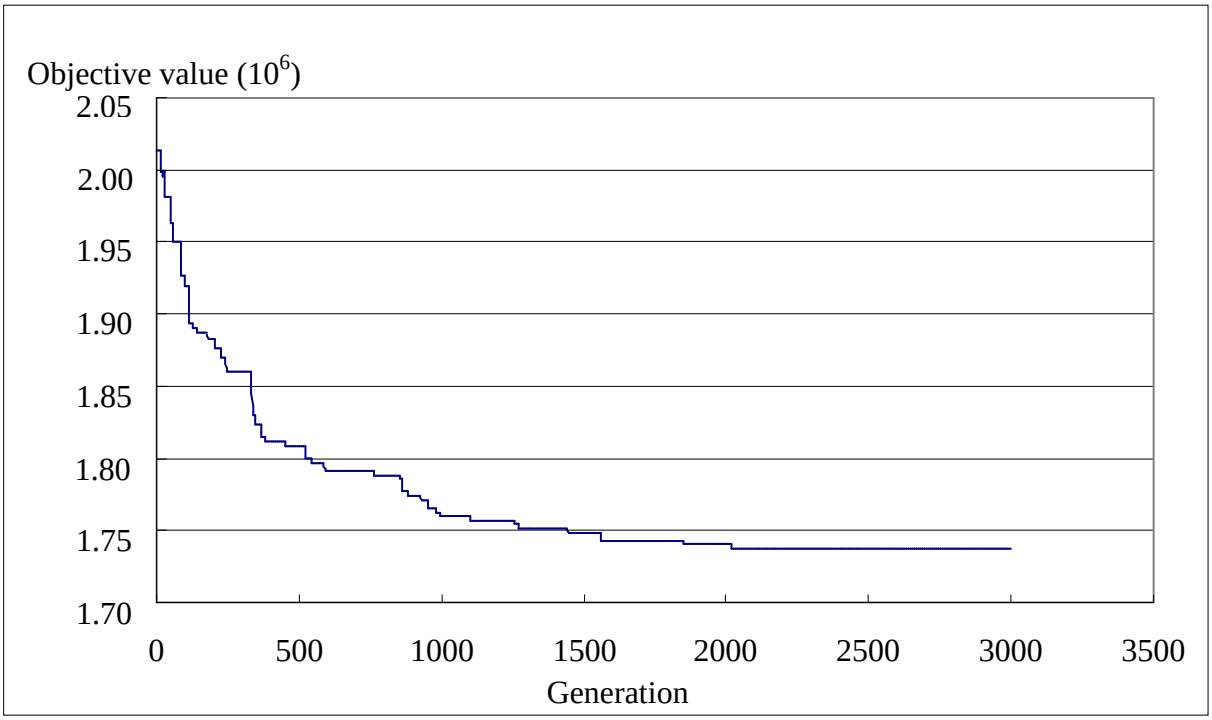


Figure 9 Convergence process of the algorithm

4.2 Effect of weights

Due to the multi-objective nature of the formulation, the settings of the weights B_1 and B_2 are important for achieving a good balance between the number of transfers and the total passengers' travel time. Replicates of the algorithm with different weights were run. Each replicate was run only once and terminated until 3000 generations were generated. In these experiments, the B_2 value was kept to be 1, and the B_1 value was varied from 0 to 200.

Figure 10 shows the computational results of these experiments. From the figure, it can be seen that the two objectives, i.e., the number of transfers and the total travel time, are generally conflicting. For the first GA replicate, B_1 was set to 0, meaning that the algorithm does not consider the number of transfers at all. The algorithm gives the largest number of transfers (9973) and smallest total travel time (1082325 minutes) among all replicates. Comparing to the existing design with the number of transfers of 6966 and the total travel time of 1578830 minutes, the number of transfers is increased by 43.2%, while the total travel time is reduced by 31.4%. The significant increase on the number of transfers renders the solution not acceptable. With B_1 increasing and B_2 unchanged, the number of transfers increases and the total travel time decreases at the same time. When B_1 was increased to 200, the algorithm achieves the smallest number of transfers (3923) and the largest total travel time (1642853 minutes). Comparing to the current design, although the number of transfers is

reduced significantly (by 43.7%), the total travel time is increased by 4.1%. Figure 9 also shows that using the B_1 values ranging from 40 to 160 can reduce both the number of transfers and the total travel time. In the following experiments, B_1 is set to 80.

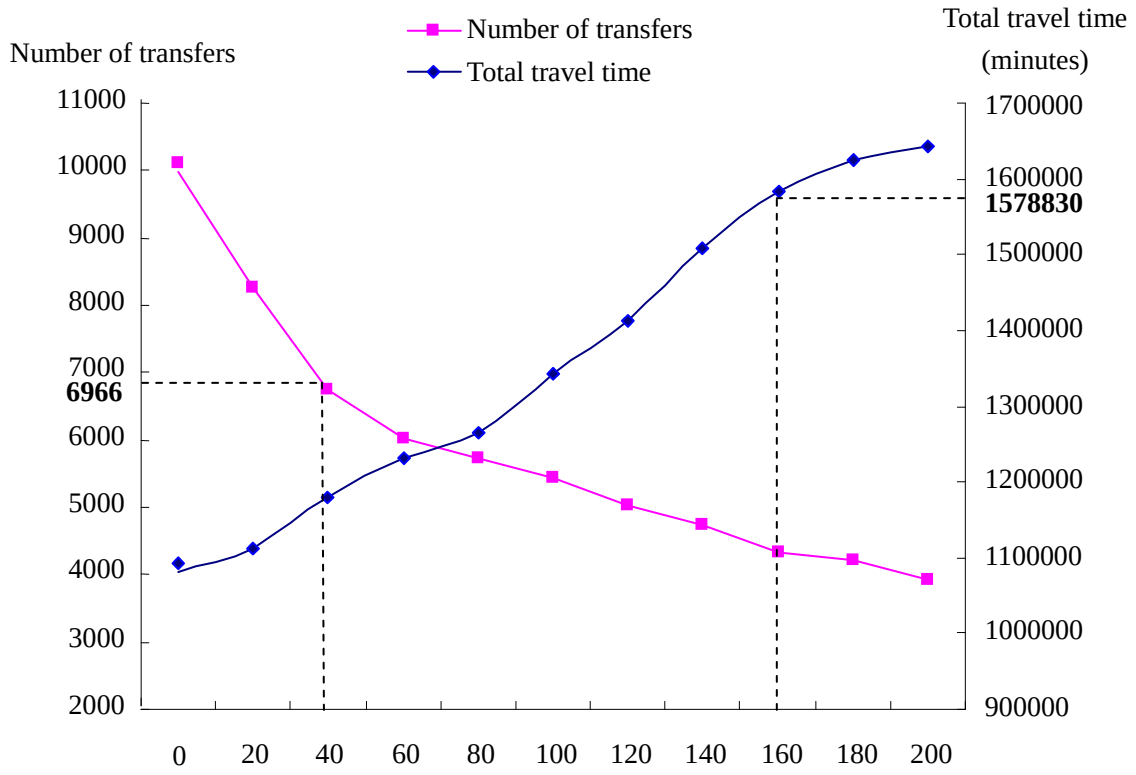


Figure 10 Effects of weights B_1 and B_2

4.3 The effect of diversity control

This example is to compare the solutions obtained from the GA with and without diversity control. Through trial and error, the values of c and a in equation (16) are set to be 0.08 and 0.002, respectively. Each GA was run for 2000 generations. Figure 11 illustrates the average hamming distance for the 2000 generations. It can be seen that for the GA without diversity control, the average hamming distance decreases rapidly in the first 1000 generations whereas for the GA with diversity control, the average hamming distance does not change much over generation.

The GAs with and without diversity control were both run 100 times for the investigated problem and the results were compared and shown in Table 2. It can be seen that the GA with diversity control outperforms the GA without diversity control in solving the problem. Comparing to the GA without diversity control, the GA with diversity control can reduce the number of transfers, the total travel time, the weighted objective value by 5.05%, 4.39%, and

4.56%, respectively. A *t*-test was also conducted to examine whether the differences are statistically significant. The *t*-test results show that the differences are all significant with a practically zero probability tail. These results indicate that the diversity control mechanism can significantly improve the performance of the GA in solving the investigated problem.

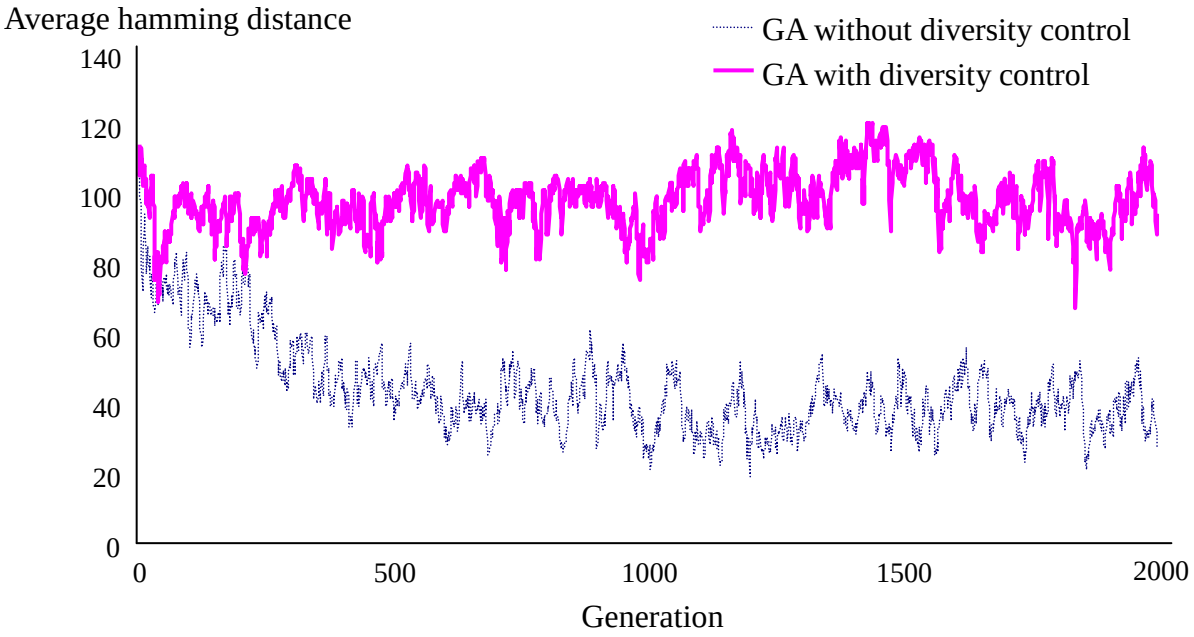


Figure 11 Average hamming distance over generations

Table 2 A comparison of GAs with and without diversity control

	Weighted Objective		Number of transfers		Total travel time	
	Average	Std. dev.	Average	Std. dev.	Average	Std. dev.
GA without diversity control	1783436.36	31740.20	5938.24	245.82	1308377.16	39146.68
GA with diversity control	1702037.76	25904.75	5638.45	186.74	1250961.76	29942.43
Improvement	4.56%	\	5.05%	\	4.39%	\

[a]Average weighted objective value for 100 runs.
[b]Standard deviation of weighted objective values of 100 runs.

4.4 Comparison of the sequential approach and the simultaneous approach

Another experiment was conducted by solving the route design problem and the frequency setting problem in a sequential manner. First, the route design problem was solved by the GA with the objective to minimize the number of transfers only (because the total travel time cannot be known without the frequency information). After that, with the route structure obtained by the GA, the frequency setting heuristics was run once to optimize the total travel time. Due to the isolation of the route design and frequency setting problems, the

total computational time was as short as 13 seconds, which is about 1/20 of that time required to solve the route design problem and frequency setting problem simultaneously. Although the computational time and complexity are much reduced, the solution obtained is not satisfactory. The number of transfers is 3733 and total travel time is 1666721 minutes. Considering the current total travel demand of 21751 passengers with the number of transfers of 6966, the percentage of passengers who need to transfer is reduced from 32% to 17%. This is equivalent to an increase of 15% of passengers who do not need to transfer. However, the total travel time is increased by 87891 minutes, which is equivalent to an increase of travel time of 4.04 minutes for every passenger on average. The result is not satisfactory because the direct service for the additional 15% of users is obtained at the expense of 4.04-minute longer travel time for every user on average. This experiment illustrates that solving the route design problem and frequency setting problem in a sequential manner does not necessarily result in an overall better solution than the existing design. This contrasts to the proposed simultaneous approach that can generate a better solution than the existing one (see figure 10) once the parameter values are properly chosen.

4.5 Comparison with the current design

In order to verify the effectiveness of the proposed method, the solution obtained was compared with the existing design. In the current design, there are 10 bus routes in operation (listed in Table 3). The total number of transfers is 6966, while the total travel time is 1578830 minutes. For all the 100 algorithm runs performed in Section 4.3, both the number of transfers and the total travel time are smaller than those for the existing design.

Table 3 Existing routes and bus allocation in the network

Routes	Stop sequence	Number of buses	Headway (minute)	In-TSW time
1	20, 19, T, 25	12	10.1	10.2
2	16, 17, 18, 23, 22, 21, T, 25	17	8.3	20.0
3	1, 6, 9, 10, 12, 13, 19, 21, T, 25	19	8.7	32.1
4	14, 13, 12, 10, 8, 16, 17, 18, 23, 22, T, 26	18	10.9	40.3
5	1, 6, 8, 16, 17, 18, 23, 22, 21, T, 28	30	4.2	33.7
6	9, 10, 11, 5, 6, 8, 16, 17, 18, 23, 22, T, 27	16	11.5	42.0
7	16, 17, 18, 23, 22, 21, T, 24	19	5.1	20.0
8	7, 6, 1, 2, 3, 4, 11, 12, 13, 19, T, 24	11	12.3	38.9
9	1, 6, 5, 4, 11, 12, 13, 19, T, 24	23	5.3	32.3
10	14, 15, 8, 9, 10, 12, 13, 19, T, 24	11	11.1	32.6

Table 4 shows the detailed route structure of the best solution among the 100 runs. For the solution, the number of transfers is 5508, which is 20.9% smaller than that of the current design, and the total travel time is 1220128 minutes, which is 22.7% smaller than that of the current design. This is equivalent to an average travel time reduction of 16.5 minutes for every user. In addition, the numbers of intermediate stops of the routes are generally reduced. The maximum number of intermediate stops is reduced from 11 to 9. The maximum travel time within TSW district is also reduced from 42.0 to 34.1 minutes.

Table 4 The best solution obtained by the proposed method

Routes	Stop sequence	Number of buses	Headway (minutes)	In-TSW time
1	1, 2, 3, 10, 15, 14, 19, T, 24	11	10.9	31.3
2	1, 5, 6, 7, 8, 16, 23, T, 25	20	8.3	33.0
3	9, 1, 5, 4, 11, 12, 13, T, 24	13	9.5	33.1
4	1, 5, 6, 7, 8, 16, 17, 19, T, 26	21	8.5	32.0
5	7, 6, 9, 15, 14, 10, 12, 13, T, 28	14	9.0	34.1
6	1, 5, 6, 7, 9, 15, 16, 17, T, 27	29	5.8	33.8
7	9, 15, 14, 21, T, 26	16	9.5	18.2
8	9, 6, 7, 8, 16, 23, 22, T, 24	16	7.3	30.3
9	1, 7, 9, 15, 14, 20, T, 25	17	9.8	32.4
10	1, 5, 7, 8, 16, 18, 23, T, 28	19	6.5	33.0

4.6 Robustness of the obtained solution

The travel demands of the network are estimated and the real demands may vary from day to day. To evaluate the robustness of the solution obtained by the proposed design, 1000 demand matrices are generated by perturbing the estimated demand matrix and used for the evaluation. For each perturbed demand matrix, the demand from node i to destination j, d'_{ij}, is randomly generated from a uniform distribution [0.9 d_{ij}, 1.1 d_{ij}], where d_{ij} is the demand of the estimated matrix.

Table 5 compares the proposed, existing, and sequential (mentioned in Section 4.4) designs using the perturbed demand matrices. It can be seen that the three designs give very different (weighted) objective values in general. Moreover, in terms of average weighted objective value, the existing design is the worse and the proposed design is the best. This conclusion has been confirmed by the results of the t-test on the differences of average weighted objective values shown in Table 6, which indicate that all the differences are statistically significant.

Table 5 A comparison of existing, sequential, and proposed designs under perturbed demands

Design	Weighted objective value		Number of transfers		Total travel time	
	Average[a]	Std. dev.[b]	Average[a]	Std. dev.[b]	Average[a]	Std. dev.[b]
Existing	2129564.03	14414.94	6941.03	69.50	1574282.28	10838.17
Sequential	1958784.36	12677.14	3713.50	43.16	1661831.48	10826.12
Proposed	1655690.64	10238.57	5485.49	56.34	1216833.55	8277.29

[a]Average weighted objective value for 1000 runs.
[b]Standard deviation of weighted objective values of 1000 runs.

Table 6 t-tests on the difference of average objective values on perturbed demands

Test	Difference of mean	*t*-value	Probability
Existing - proposed	473873	846.75	0.000
Sequential - proposed	303094	587.62	0.000

5 Conclusions

A trunk bus network design problem for a suburban residential area in Hong Kong has been investigated. The problem aims to reduce the number of transfers and the total travel time of the network. Because these two objectives are conflicting and the latter is determined by both the route design and the frequency setting problems, the route design problem and frequency setting problem is considered simultaneously. The analysis of the established integrated mathematical model points out that the problem is a mixed integer programming problem and a heuristic method is required to solve the problem efficiently.

An integrated solution method is proposed to solve the route design and frequency setting problems simultaneously. The solution method integrates a specific genetic algorithm which aims to optimize the route design, and a neighborhood search heuristic which aims to optimize the frequency setting. A new solution representation scheme is proposed, which enables the genetic algorithm to search within all possible route structures. To improve the effectiveness of the solution process, two crossover (route crossover and stop crossover) and four mutation operators (insert, remove, swap, and transfer) are developed. Experiments show that incorporating all these operators in the algorithm is beneficial for improving the performance of the algorithm.

Through experiments, we find that the setting of the weights is important for achieving a good balance between the two objectives, i.e., minimizing the number of transfers and minimizing total travel time. Although the best balance between the two objectives depends on the discretion of the operator or the public who may concern, there is a wide range for the parameters that can lead to a design better than the current design in terms of both objective

measures.

A diversity control mechanism is proposed to improve the performance of the GA in solving the investigated problem. Experiments show that the GA with the diversity control mechanism outperforms the GA without the diversity control mechanism in terms of lower number of transfers, smaller total travel time, and smaller weighted objective values. The differences are statistically significant.

To illustrate the robustness and quality of solutions obtained, computational experiments are performed based on 1000 perturbed demand matrices. The t-test results show that the design obtained by the proposed solution method is robust under demand uncertainty, and the design is better than both the current design and the design obtained by solving the route design problem and the frequency setting problem sequentially. Compared with the current bus network design, the proposed method can generate a design which can reduce both the number of transfers and total travel time at least by 20.9% and 22.7% respectively.

This study opens many future research directions. One example is that the study can be extended to consider bus service timetables as in Zhao and Zeng (2008), multi-period as in Marína and Jaramillob (2008), clean bus allocation as in Beltran et al. (2009), and limited-stop services as in Leiva et al. (2010). Another example is that other meta-heuristics can be developed for the studied problem and compared with the proposed solution method. One can also incorporate existing route construction algorithm (e.g., Mauttone and Urquhart, 2009) in our proposed solution method and study the improvement of solution quality and computation time. Finally, extending the proposed solution methodology to solve other transport network design problems (e.g., Part et al. 2009 and Chen et al. 2010) can be another future research direction.

Acknowledgement

The research was supported by a grant (201001159008) from the University Research Committee of the University of Hong Kong. The authors are grateful for the two anonymous referees for their constructive comments.

References

Baaj H, Mahmassani HS. TRUST: A LISP Program for the analysis of transit route configurations. *Transportation Research Record* 1990; 1283; 125–135.

Bel G, Dubois D, Llibre M. A set of methods in transportation network analysis and synthesis.

International Journal of Operational Research Society 1979; 30 (9); 797–808.

Beltran B, Carrese S, Cipriani E, Petrelli M. Transit network design with allocation of green vehicles: A genetic algorithm approach. *Transportation Research Part C* 2009; 17 (5); 475-483.

Bielli M, Caramia M, Carotenuto P. Genetic algorithms in bus network optimization. *Transportation Research Part C* 2002; 10 (1); 19–34.

Borndöfer R, Gröschel M, Pfetsch ME. A path-based model for line planning in public transport. Tech. Rep. Report 05-18, ZIB; 2005.

Bussieck MR. *Optimal Lines in Public Rail Transport*. Ph.D. Thesis, TU Braunschweig; 1998.

Carrese S, Gori S. An urban bus network design procedure. *Transportation Planning: State of the Art*. (Eds. Michael Patriksson and Martine Labbé) 2002; 177–196. Kluwer Academic Publishers

Ceder A, Wilson NHM. Bus network design. *Transportation Research Part B* 1986; 20 (4); 331–344.

Chen, A., Pravinvongvuth, S. and Chootinan, P. Scenario-based multi-objective AVI reader location models under different travel demand patterns. *Transportmetrica* 2010; 6 (1), 53-78.

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

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

Fusco G, Gori S, Petrelli M. An heuristic transit network design algorithm for medium size towns. *The Proceedings of the 13th Mini-EURO Conference*, Bari, Italy; 2002.

Guihaire V, Hao JK. Transit network design and scheduling: A global review. *Transportation Research Part A* 2008; 42 (10); 1251–1273.

Kuan SN, Ong HL, Ng KM. Solving the feeder bus network design problem by genetic algorithms and ant colony optimization. *Advances in Engineering Software* 2006; 37 (6); 351–359.

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

Lee YJ, Vuchic VR. Transit network design with variable demand. *Journal of Transportation*

Engineering 2005; 131 (1); 1–10.

Leiva C, Muñoz JC, Giesen R., Larrain H. Design of limited-stop services for an urban bus corridor with capacity constraints *Transportation Research Part B* 2010; in press.

Magnanti TL, Wong RT. Network design and transportation planning: models and algorithms. *Transportation Science* 1984; 18 (1); 1–55.

Marína Á, Jaramillob P. Urban rapid transit network capacity expansion. *European Journal of Operational Research* 2008; 191 (1); 45–60.

Martins CL, Pato MV, Search strategies for the feeder bus network design problem. *European Journal of Operational Research, Volume* 1998; 106 (2-3); 425–440.

Mauttone A, Urquhart ME. A route set construction algorithm for the transit network design problem. *Computers & Operations Research*, 2009; 36 (8); 2440–2449.

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

Pacheco J, Alvarez A, Casado S, González-Velarde JL. A tabu search approach to an urban transport problem in northern Spain. *Computers & Operations Research* 2009; 36 (3); 967–979.

Park, B., Yun, I. and Ahn, K. Stochastic optimization for sustainable traffic signal control. *International Journal of Sustainable Transportation* 2009; 3 (4), 263-284.

Shih M, Mahmassani HS. A design methodology for bus transit networks with coordinated operations. Tech. Rep. SWUTC/94/60016-1, Center for Transportation Research, University of Texas, Austin; 1994.

Shih M, Mahmassani HS, Baaj M. A planning and design model for transit route networks with coordinated operations. *Transportation Research Record* 1998; 1623; 16–23.

Shimodaira H. A Diversity-Control-Oriented Genetic Algorithm (DCGA): Development and experimental results. *Proceedings of the Genetic and Evolutionary Computation Conference* (*GECCO-99*) *Volume I*, Morgan Kaufmann, 1999, pp.603–611.

Shimodaira H. DCGA: A diversity control oriented genetic algorithm. *Proceedings of the Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Glasgow, UK, 1997, pp. 444–449.

Shimodaira H. A Diversity-Control-Oriented Genetic Algorithm (DCGA): Performance in function optimization. *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, Korea, 2001, pp. 44–51.

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

Van Nes R, Hamerslag R, Immers BH. Design of public transport networks. *Transportation Research Record* 1988; 1202; 74–83.

Wan QK, Lo HK. A mixed integer formulation for multiple-route transit network design. *Journal of Mathematical Modelling and Algorithms* 2003; 2 (4); 299–308.

Zhao F, Zeng XG. Optimization of user and operator cost for large-scale transit network. *Journal of Transportation Engineering* 2007; 133 (4); 240–251.

Zhao F, Zeng XG. Optimization of transit route network, vehicle headways and timetables for large-scale transit networks. *European Journal of Operational Research* 2008; 186 (2); 841–855.

Appendix I Travel demands of the network

Node	24	25	26	27	28	Total
1	595	372	256	192	441	1856
2	167	99	97	46	162	571
3	147	100	97	57	164	565
4	106	57	53	29	92	337
5	313	187	196	114	301	1111
6	270	194	179	95	336	1074
7	349	192	178	94	307	1120
8	313	191	179	97	345	1125
9	298	158	124	71	254	905
10	104	62	48	33	101	348
11	60	36	35	21	70	222
12	485	337	287	142	488	1739
13	547	264	226	158	423	1618
14	196	120	92	62	177	647
15	316	203	158	82	276	1035
16	784	425	375	258	629	2471
17	87	52	50	30	81	300
18	237	158	147	79	211	832
19	107	63	57	30	90	347
20	186	98	77	55	147	563
21	113	58	57	32	83	343
22	104	63	51	35	87	340
23	638	461	369	197	617	2282
Total	6522	3950	3388	2009	5882	21751

Note: 1. The demands are from 23 TSW nodes to 5 destinations in the city
2. The demands are hourly demands during peak hours