Hybrid Evolutionary Metaheuristics for Concurrent Multi-Objective Design of Urban Road and Public Transit Networks

Elnaz Miandoabchi[1], Reza Zanjirani Farahani[1*], Wout Dullaert[2,3], Wai Yuen Szeto[4]

(1) Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

(2) Institute of Transport and Maritime Management Antwerp, Antwerp, Belgium

(3) Antwerp Maritime Academy, Antwerp, Belgium

(4) Department of Civil Engineering, The University of Hong Kong, Hong Kong, China

Abstract

This paper addresses a bi-modal multi-objective discrete urban road network design problem. The problem includes the simultaneous design of urban road and bus networks in urban areas in which the authorities play a major role in designing bus network topology. The road network design consists of decision making for lane additions to the existing streets, new street constructions, determining the orientations and locations of one-way streets, and lane allocations for two-way streets. The bus network design is performed by keeping the terminal stations of the existing bus lines unchanged and redesigning the forth and back routes of each line. Four measures, namely user benefit, the demand share of the bus mode, the demand coverage of the bus network, and the average travel generalized cost of bus passengers, are used to evaluate the network design scenarios. The interaction of automobile and bus flows is explicitly modeled using a modal-split/assignment model to obtain the flows in the deterministic user equilibrium state. The problem is formulated as a mathematical program with equilibrium constraints. A hybrid of genetic algorithm and simulated annealing as well as a hybrid of clonal selection algorithm and simulated annealing are proposed to solve the problem. Computational results for a number of test instances are presented and investigated.

Keywords: Bimodal; Transit network design, Road network design, Multi-objective; Elastic demand; Hybrid metaheuristics

1. Introduction

One of the most challenging issues in which authorities in urban areas always face with is to cope with the ongoing growth of travel demand and increasing volumes of traffic in urban transportation networks. A wide range of network improvement plans such as expansion and reconfiguration of

* Corresponding. Fax: +98 21 66413025; Tel: +98 21 66413034, +98 21 66466497;Email: zanjiranireza@gmail.com

streets, adjustment of traffic control systems and road tolls, improvement of public transit systems, etc. are usually implemented in urban transportation networks to cope with the problem. These network improvement plans can be studied and analyzed under the branch of "Urban Transportation Network Design Problems" (UTNDPs) that deal with the designs of road or public transit networks. These problems can be further divided into four categories, namely the "Road Network Design Problem" (RNDP), the "Transit Network Design Problem" (TNDP), the "Bimodal Network Design Problem" (BMNDP), and the "Multi-Modal Network Design Problem" (MMNDP).

The RNDP is often called "Network Design Problem" (NDP) in the literature. The problem is modeled as Stackelberg leader-follower type game. Government as the network authority plays the role of the leader who decides to change the network design, and network travelers react with the decision of the leader and determine their travel routes. Besides, travelers compete with each other to choose the required routes. As a result, the problem can be expressed as a bi-level programming formulation, where the upper level problem simulates the decision making of the leader based on the existing constraints and specific criteria, and the lower level problem simulates the response of the traveler community to the leader's decisions. The lower level problem itself is modeled using the Nash-Cournot equilibrium concept to depict the competition of the travelers.

The RNDP generally considers all vehicles such as private cars, taxies or even buses as a single mode. These problems are considered in two forms: Discrete NDP (DNDP) and Continuous NDP (CNDP). The DNDP deals with discrete decision variables such as the construction of new streets and defining the orientation of the streets, and the CNDP is concerned with continuous network attributes such as setting signal timing and toll. The Mixed Network Design Problem (MNDP) is another less investigated form that combines both discrete and continuous decision variables. Comprehensive reviews on the RNDPs can be found in Magnanti and Wong (1984), Friesz (1985), Migdalas (1995), and Yang and Bell (1998).

The TNDP deals with the design of public transportation networks in terms of network attributes and the operational conditions of the public transit vehicles. Every public transportation network is composed of a set of transit routes which are defined by predetermined sequences of stations between two terminals. A group of transit vehicles such as buses, metro, trains, trams, etc., operates based on a predefined schedule on each route. A significant body of research exists in the field of the TNDP, and a large portion of it is directed towards bus network design. The hierarchy of bus network design decisions in the TNDP can be presented as strategic, tactical, and operational levels (Ceder and Wilson, 1986): network or topology design (strategic level), frequencies setting, timetable development (both tactical), bus scheduling, and driver scheduling (both operational). These problems are studied individually or in combination with each other, and a wide spectrum of related works exists as the result of different assumptions, modeling and solution approaches. The use of bi-level programming such as in the RNDP is not common in all TNDPs; for example network topology design problems are often modeled as single level problems assuming shortest path selection or other simple user route selection behaviors. On the other hand, equilibrium route choice behavior (bi-level programming) is mainly used when frequency setting is the sole or one of the decisions of the problem. Desaulniers and Hickman (2007) and Guihaire and Hao (2008) provide a comprehensive review of the TNDPs.

RNDPs and TNDPs usually assume fixed demand share and also do not consider the impact on the decisions on the mode under focus on the other existing modes; while the decisions of road network might affect the performance and demand share of the public transportation network and vice versa. The interaction of automobile and bus flows is considered in a few studies in TNDPs (e.g., Fernández *et al.*, 2008) and RNDPs (e.g., Cantarella and Vitetta. 2006).

The BMNDP is concerned with both public transit and private automobile modes, while the MMNDP considers more than two modes. In contrast to most of RNDPs and TNDPs, in these problems demand for each mode is elastic, and also the effects of decisions of one mode on the performance and demand share of the other modes is taken into account. Studies in the MMNDP and BMNDP focus on both general cases (e.g., Van Nes, 2002; Clegg *et al.*, 2001) and specific problems such as pricing problems (e.g., Bellei *et al.*, 2002; Ying and Yang, 2005; D'Acierno *et al.*, 2006; Hamdouch *et al.*, 2007), the transit fare or frequency setting problem (e.g., Lo *et al.*, 2003; D'Acierno *et al.*, 2005; Uchida *et al.*, 2007), and the simultaneous road toll, ticket price, and transit frequency setting problem (e.g., Ferrari, 1999).

As can be seen from the literature, TNDPs ignore the existing road network restrictions and at most consider a set of two-way streets network as the base for the design of bus network. In realty one-way streets may cause complications on TNDPs as the forth and back routes of a bus line cannot be on the same one-way street. On the other hand, RNDPs and especially those with street orientation decisions do not consider the interferences with the existing bus network, and the decisions may result in a disconnected bus network.

Although usually private companies operate bus lines, there are cases all over the world that public authority operates the bus lines and that the private bus line operators have the least authority to decide to modify the existing bus routes in the network. In such cases, where public authority has the main role in determining the bus network design, if the redesign of the bus network is possible when deciding to invest on road network improvements, advantages can be gained by improving both networks simultaneously. The decisions of both networks can be made jointly to improve the performances of both networks, leading to a more efficient bus network and a less congested road network.

The problem investigated in this paper is a BMNDP which encompasses both road network design and public transit network topology design decisions. The decisions included in the BMNDP addressed in this paper are as follows:

a. Constructing new streets;
b. Adding new lanes to the existing network streets;
c. Determining the directions and locations of one-way streets;
d. Determining lane allocations in two-way streets, and
e. Determining the topology of the bus network.

Items (a) to (d) are related to discrete road network design decisions. These decisions have been studied in many works, individually or in combination with each other. New street construction has been investigated as an individual decision and sometimes in combination with lane additions (e.g., Steenbrink, 1974; LeBlanc, 1975; Poorzahedy and Turnquist, 1982; Poorzahedy and Abulghasemi, 2005; Poorzahedy and Rouhani, 2007). Street capacity expansion has usually been studied as a continuous decision variable, as a huge body of research in the RNDP belongs to the CNDP with street expansion (e.g., Dantzig *et al*. 1979; Patil and Ukkusuri, 2007; Ukkusuri *et al*. 2007). Discrete capacity expansion in the form of lane addition has also been studied in the literature of the DNDP. Lane addition has been considered in the studies in the form of do not-do decision, among other decisions such as resurfacing of roads or improving the capacity of existing street lanes (e.g., LeBlanc, 1975; Jeon *et al.* 2006). The street orientation problems in either all one-way or mixed one-way and two-way form have been studied by some authors (e.g., Roberts and Xu, 1988; 1989; 1992; 1994; Lee and Yang, 1994; Drezner and Wesolowsky, 1997; Drezner and Salhi, 2000; 2002). The orientation of streets has been considered in combination with signal timings in Cantarella *et al.* (2006) and Gallo *et al.* (2010), and in combination with street construction in Drezner and Wesolowsky (2003). Reallocations of street lanes in peak hours has been considered in Zhang and Gao (2007), and the simultaneous considerations of lane allocations, street orientations, signal timings and parking lane allocations have been investigated by Cantarella and Vitetta (2006).

In the previous works of the authors, a multi-objective DNDP (Miandoabchi *et al.*, 2010), and a multi-objective BMNDP (Miandoabchi *et al.*, 2011) were proposed. The former work considers decisions (a) to (d) in two problem variations in terms of two-way streets lane allocations. The latter considers decision set (a)-(d) together with allocation of exclusive bus lanes, assuming a fixed existing bus network. The present work involves bus network topology design as the fifth decision. The inclusion of the latter transforms the problem to a completely new BMNDP which has been never investigated before. The simultaneous design of bus and road networks requires a compromise between the performances of both networks. As the result, the problem is formulated as a multi-objective mixed-integer nonlinear bi-level programming model with a combined modal-split/assignment problem in its lower level. Because of the high degree of the complexity and non-convexity of the model, two multi-objective hybrid evolutionary algorithms are proposed to solve the model. Both algorithms attempt to obtain a set of Pareto-optimal solutions.

Indeed, the multi-objective network design problem has been already under focus in the UTNDP. The multi-objective RNDP has been studied in a number of research works (e.g., Friesz *et al.*, 1993; Yang and Wang, 2002; Ukkusuri *et al.*, 2007; Chiou, 2008; Szeto *et al.*, 2010). TNDPs are intrinsically multi-objective problems and a variety of user and operator related objective functions can be found in the literature (see Guihaire and Hao (2008) for the review of the objective functions). The contribution of this paper can be summarized as follows:

1. A new multi-objective, bi-modal network design problem is proposed which includes both the bus network topology design and DNDP decision, and explicitly considers the interactions between bus and automobile flows.
2. A new hybrid meta-heuristic for the problem is proposed.

The rest of the paper is organized as follows: In the next section the problem is defined. In section 3, the notations and mathematical formulation of the problem are given. In section 4, the metaheuristic solution procedures are proposed to solve the network design problem under focus. Section 5 contains the computational results. Finally, conclusions and future research suggestions are made in section 6.

2. Problem Definition

The problem under consideration is to design a bimodal urban transportation network under a multi-objective decision-making framework taking into account the effects of automobile flows on bus flows or vice versa. All types of streets in road networks are called "link". Each two-way link consists of two arcs and each one way link consists of one arc. Each arc has a number of lanes allocated to it. If a movement is not allowed in one direction of a link, no arc will exist in that direction.

A bus network is defined by a set of "bus lines" which operate on their specified routes. In this paper, the design of the bus network is carried out by keeping the terminal stations unchanged and redesigning the forth and back routes of each existing bus line. A bus line is identified by two sequences of road network nodes, as the forth and back routes between two terminal stations, in which a group of buses operate on it. The forth and back routes may or may not pass the same sequence of nodes. Arcs of the road network can be distinguished as two types: (1) arcs that bus lines pass through them are referred to as "bimodal arcs", and (2) arcs that no bus lines pass through them are referred to as "auto arcs".

2.1. Assumptions

The main assumptions for the studied problem are listed below:

- A basic road network with all two-way links exists in advance;
- A previous bus network exists in which it is intended to be redesigned by keeping the existing bus lines, but modifying their forth and back routes;
- Links are allowed to have different numbers of lanes in each direction;
- Travel demand between each origin and destination (OD) pair is fixed and known, but the travel demand for each mode is elastic and is determined by a logit modal split function;
- Auto and bus flows influence each other, and thus the travel cost function for each mode is asymmetric;
- The forth and back routes of a bus line are not restricted to pass through the same set of nodes;
- Automobile drivers follow the deterministic user equilibrium principle in selecting their paths whereas bus passengers select bus lines based on the shortest hyperpath (e.g., Nguyen *et al.*, 1998; Spiess and Florian , 1989; De Cea and Fernández, 1993) and;
- Automobile flows and bus passenger flows are determined by solving the combined modal-split/traffic assignment problem.

2.2. Inputs

Having described the problem and assumptions, the following are the required inputs:

- The estimated total travel demand matrix;
- The terminal node pair and frequency for each bus line;
- Maximum allowable round trip time for each bus line;
- Attributes of network links such as capacities, investment costs, and travel cost functions;
- Candidate new link construction projects and their costs (a link construction project includes construction of a new street between two nodes of the network). The candidate link construction projects consist of a set of possible street constructions that has been identified by the network authority;
- Maximum allowable number of lane additions to network links and their costs (a lane addition project includes construction of a pair/s of new lanes on each side of the existing street). This number is determined by the network authority;
- Budget limit for lane addition and link construction projects, and
- The values of walking, waiting and travel times, auto toll, and bus fare.

2.3. Outputs

The following outputs are obtained from our proposed model:

- The orientation and locations of one-way links;
- The number of lanes allocated to each direction of each network link;
- New link constructions;
- Lane additions to the existing network links, and
- Forth and back routes of each bus line.

3. Mathematical models and notations

In this section, the mathematical model and its notations are described, and then the specific functions used in problem formulation are presented. Let the directed graph G (N, A) be the existing network, where N is the set of intersection nodes, origin nodes, and destination nodes, and A the set of existing network arcs. In such a network, an automobile and a basic bus network can be distinguished. The former corresponds to the road network and the latter is defined by a set of bus lines passing the road network arcs. Each bus line is defined by a set of binary variables, stating the arcs that the bus line passes through in its forth and back routes between the two bus terminals.

Since this study focuses on bus network planning rather than actual bus operation, the formulation proposed must be able to handle large bus networks. Therefore, the frequency-based approach is more suitable than the schedule-based approach (e.g., Nuzzolo *et al*., 2001) to model passenger assignment. Among the existing frequency-based modeling frameworks, the modeling framework proposed by Uchida *et al.* (2007), which is derived from De Cea and Fernández (1993), can handle larger transit networks compared with those proposed by Nguyen *et al.* (1998) and Spiess and Florian (1989) because some of line sections in the bus network are combined. Moreover, Uchida *et al.* (2007) can handle the interaction between multiple modes and the in-vehicle congestion effect whereas Nguyen *et al.* (1998) and Spiess and Florian (1989) cannot. Therefore, the modeling framework of Uchida *et al*. (2007) is used with some modifications to account for different road network design decisions.

In the modeling approach of Uchida *et al.* (2007), the basic bus network is transformed into a modified bus network before combining with the automobile network to form a hyper-network. Since in our problem the bus routes are designed for each network design scenario $\mathbf{y}$, a modified network is created for each scenario. Each modified network is denoted by $\hat{G}\left(\hat{N},\hat{A}(\mathbf{y})\right)$ where $\hat{N}$ is the subset of road network nodes, and $\hat{A}(\mathbf{y})$ the set of the route sections and walking links. A route section is defined by a portion of a route between two consecutive transfer nodes and is associated with a set of attractive lines. The set of attractive lines for each route section is obtained by solving the common line minimization problem (see De Cea and Fernández, 1993; Uchida *et al*., 2007). Walking links are used to capture the effect of walking time on travel choice in our proposed model and connect bus stops to Origin/Destination nodes as well as these nodes to the parking locations of the expanded automobile network where candidate new links are included. To simplify the hyper-network, walking time in the automobile network is assumed to be negligible in this network, and bus stops are assumed near the head or tail of the arc. The first assumption is justifiable as the walking distance from the origin to the parking location is short, and only the difference of the walking time to the bus stop and that to the parking location is important to determine the mode choice. The second assumption is not too unrealistic since there are some real world examples such as the city of West Lafayette to support this assumption.

3.1. Sets

The following notations are used in the model formulation.

N: set of network nodes i, j, p, q, p', q'

$\hat{N}$: set of modified bus network nodes $\hat{i}, \hat{j}$

A: set of existing network arcs (i, j)

A': set of candidate new network arcs (i, j)

$\hat{A}(\mathbf{y})$: set of modified bus network arcs $(\hat{i}, \hat{j})$ for a specific design scenario $\mathbf{y}$

$\tilde{A}_{\hat{i}\hat{j}}(\mathbf{y})$: set of road network arcs which affect the travel time on modified bus network arc $(\hat{i}, \hat{j})$ for a specific design scenario $\mathbf{y}$

L: set of existing network links l

L': set of candidate new network links l'

S_l : set of arcs corresponding to existing network link l

$S_{l'}$: set of arcs corresponding to candidate new network link l'

W: set of all OD pairs (p, q)

E: set of bus lines e

$\hat{E}_{\hat{i}\hat{j}}(\mathbf{y})$: set of bus lines passing through modified bus network arc $(\hat{i}, \hat{j})$ for a specific design scenario $\mathbf{y}$

$\bar{E}_{\hat{i}\hat{j}}(\mathbf{y})$: set of attractive bus lines passing through modified bus network arc $(\hat{i}, \hat{j})$ for a specific design scenario $\mathbf{y}$

R_{pq}: set of auto routes r between OD pair (p, q)

$\hat{R}_{pq}(\mathbf{y})$: set of bus routes $\hat{r}$ between OD pair (p, q) for a specific design scenario $\mathbf{y}$

T_e: set of terminal node pairs corresponding to bus line e

$\hat{W}(\mathbf{y})$: set of OD pairs that are connected by the bus network ($\hat{W} \subset W$) for a specific design scenario $\mathbf{y}$

3.2. Decision Variables

y_l: number of lanes added to existing link l

$u_{l'}$: binary variable, which equals 1 if link l' is built, and zero otherwise

z_{ij}: binary variable, which equals 1 if arc (i, j) is built or present, and zero otherwise

k_{ij}: number of auto or shared auto-bus lanes allocated to arc (i, j)

$\lambda_{ijoo'}^{e}$: binary variable, which equals 1 if the route in bus line e which starts from terminal node o and ends in terminal node o', passes through arc (i, j) and is zero otherwise

X_r^{pq}: auto traffic flow on route $r \in R$ (vehicles per hour)

$\hat{X}_{\hat{r}}^{pq}$: bus passenger flow on route $\hat{r} \in \hat{R}$ (passengers per hour)

$\mathbf{y}$: vector of design variables representing a design scenario, $\mathbf{y} = \left[y_l, u_{l'}, z_{ij}, k_{ij}, \lambda_{ijoo'}^{e} \right]$

$\mathbf{x}$: vector of path flows, $\mathbf{x} = \left[X_r^{pq}, \hat{X}_{\hat{r}}^{pq} \right]$

3.3. Parameters

$\bar{d}_{pq}$: total travel demand between OD pair (p, q)

B: budget limit for lane addition and link construction projects

y_l^{max}: maximum allowable number of lanes added to each side of existing link l

H_e: maximum allowable round trip time of bus line e

K_l: current number of lanes on existing link l

$K_{l'}$: number of lanes on candidate new link l'

δ_{ijr}: binary variable, which equals 1 if route r uses arc (i, j), and zero otherwise

$\delta_{\hat{i}\hat{j}\hat{r}}$: binary variable, which equals 1 if route $\hat{r}$ uses arc $(\hat{i}, \hat{j})$, and zero otherwise

π: auto occupancy (average number of passengers in one auto)

M: a large positive number

ψ: bus-specific constant, $\psi \geq 0$

θ: mode choice parameter, $\theta \geq 0$

ρ_{ij}: auto toll on arc (i, j)

$\hat{\rho}_{\hat{i}\hat{j}}$: the fixed bus fare for traveling between $\hat{i}$ and $\hat{j}$

3.4. Functions of Decision Variables

w_{pq}: binary variable, which equals 1 if OD pair (p, q) are connected in the bus network or a person from p can reach q through the bus network

$\xi^{e}_{ijoo'}$: binary variable, which equals 1 if a person from node i can reach node j by bus line e with the starting terminal node o and ending terminal node o', and zero otherwise

d_{pq} : auto demand between OD pair (p, q) (passengers per hour)

$\hat{d}_{pq}$: bus passenger demand between OD pair (p, q) (passengers per hour)

x_{ij}: auto traffic flow on arc (i, j) (vehicles per hour)

$\hat{x}_{\hat{i}\hat{j}}$: bus passenger flow on arc $(\hat{i}, \hat{j})$ (passengers per hour)

$\tilde{x}_{\hat{i}\hat{j}}$: the sum of bus passenger flow boarding the lines passing through arc $(\hat{i}, \hat{j})$ before $\hat{i}$ and alighting after $\hat{i}$ and the bus passenger flow boarding at node $\hat{i}$ and alighting after $\hat{j}$ (passengers per hour)

$\chi^{e}_{\hat{i}\hat{j}}$: bus passenger flow of line e on arc $(\hat{i}, \hat{j})$ (passengers per hour)

$\tilde{\chi}^{e}_{\hat{i}\hat{j}}$: bus passenger flow competing line e with the passenger flow on arc $(\hat{i}, \hat{j})$ (passengers per hour)

Λ_{ij} : binary variable, which equals 1 if arc (i, j) is bimodal, and zero otherwise

$g_l(y_l)$: investment cost function for the expansion of existing link l, when y_l lanes are added to both sides of the link

$g'_{l'}(u_{l'})$: investment cost function for the construction of candidate new link l'

t^{1}_{ij}: travel cost function of the auto mode on arc (i, j) in the absence of bus flows

t^{2}_{ij} : travel cost function of the auto mode on arc (i, j) in the presence of bus flows

t^{3}_{ij} : travel cost function of the bus mode on arc (i, j)

$\hat{t}_{\hat{i}\hat{j}}$: in-vehicle travel cost function for bus passengers on arc $(\hat{i}, \hat{j})$

$\hat{\tau}^{e}_{\hat{i}\hat{j}}$: in-vehicle travel cost function for bus passengers of line e on arc $(\hat{i}, \hat{j})$

$t^{a}_{\hat{i}\hat{j}}$: access time cost function of bus passengers on arc $(\hat{i}, \hat{j})$

$\tilde{t}_{\hat{i}\hat{j}}$: waiting time cost function of bus passengers on arc $(\hat{i}, \hat{j})$

h_{pq} : minimum generalized travel cost of auto between OD pair (p, q)

$\hat{h}_{pq}$: minimum generalized travel cost of bus passengers between OD pair (p, q)

$\sigma_{p'q'ij}$: binary variable, which is equal to 1 if arc (i, j) is on the shortest auto path connecting OD pair (p', q') and is equal to zero otherwise

Z_1: first objective function – total user benefit

Z_2: second objective function – the demand share of the bus mode

Z_3: third objective function – the demand coverage of the bus network

Z_4: fourth objective function – average generalized travel cost of a trip by bus

3.5. Mathematical Model

The problem can be formulated as a Mathematical Programming model with Equilibrium Constraints (MPEC). Although presented as a single level model, the problem is intrinsically bi-level in which the equilibrium constraints are the lower level model of the problem. The upper level problem is formulated as a multi-objective mixed-integer nonlinear mathematical model and the lower level problem is expressed as Variational Inequality (VI) constraints. Mathematically, the problem can be formulated as follows:

$$\max \; Z_1(\mathbf{x},\mathbf{y}) = -\frac{1}{\theta} \sum_{(p,q)\in W} \bar{d}_{pq} \ln\left(e^{-\theta\,\hat{h}_{pq}} + e^{-\theta(h_{pq}-\psi)}\right) \tag{1}$$

$$\max \; Z_2(\mathbf{x},\mathbf{y}) = \frac{\sum_{(p,q)\in W} w_{pq}\hat{d}_{pq}}{\sum_{(p,q)\in W} \bar{d}_{pq}} \tag{2}$$

$$\max \; Z_3(\mathbf{x},\mathbf{y}) = \frac{\sum_{(p,q)\in W} w_{pq}\bar{d}_{pq}}{\sum_{(p,q)\in W} \bar{d}_{pq}} \tag{3}$$

$$\min \; Z_4(\mathbf{x},\mathbf{y}) = \frac{\sum_{(p,q)\in W} w_{pq}\hat{h}_{pq}}{\sum_{(p,q)\in W} w_{pq}} \tag{4}$$

Subject to

$$\sum_{l\in L} g_l(y_l) + \sum_{l'\in L'} g'_{l'}(u_{l'}) \le B \tag{5}$$

$$0 \le y_l \le y_l^{\max} \qquad \forall l \in L \tag{6}$$

$$k_{ij} + k_{ji} = K_l + 2y_l \qquad \forall l \in L, (i,j), (j,i) \in S_l \tag{7}$$

$$z_{ij} + z_{ji} \ge 1 \qquad \forall l \in L, (i,j), (j,i) \in S_l \tag{8}$$

$$k_{ij} + k_{ji} = K_{l'} u_{l'} \qquad \forall l' \in L', (i,j), (j,i) \in S_{l'} \tag{9}$$

$$z_{ij} + z_{ji} \geq u_{l'} \quad \forall l' \in L', (i,j), (j,i) \in S_{l'} \tag{10}$$

$$z_{ij} + z_{ji} \leq M\, u_{l'} \quad \forall l' \in L', (i,j), (j,i) \in S_{l'} \tag{11}$$

$$z_{ij} \leq k_{ij} \quad \forall (i,j) \in A \cup A' \tag{12}$$

$$k_{ij} \leq M z_{ij} \quad \forall (i,j) \in A \cup A' \tag{13}$$

$$\sum_{(i,j)\in A\cup A'} \sigma_{p'q'ij} [(1-\mathit{\Lambda}_{ij}) t^1_{ij} + \mathit{\Lambda}_{ij} t^2_{ij}] \leq M \quad \forall p', q' \in N \tag{14}$$

$$\mathit{\Lambda}_{ij} \leq z_{ij} \quad \forall (i,j) \in A \cup A' \tag{15}$$

$$\prod_{e\in E, (o,o')\in T_e} (1-\lambda^e_{ijoo'})(1-\lambda^e_{ijo'o}) = 1 - \mathit{\Lambda}_{ij} \quad \forall (i,j) \in A \cup A' \tag{16}$$

$$\sum_{j\in N} \lambda^e_{ojoo'} = 1 \quad \forall e \in E, (o,o') \in T_e, (o,j) \in A \cup A' \tag{17}$$

$$\sum_{i\in N} \lambda^e_{io'oo'} = 1 \quad \forall e \in E, (o,o') \in T_e, (i,o') \in A \cup A' \tag{18}$$

$$\sum_{i\neq j, i\in N} \lambda^e_{ijoo'} = \sum_{j\neq k, k\in N} \lambda^e_{jkoo'} \quad \forall e \in E,\ (o,o') \in T_e, j \in N \tag{19}$$

$$\sum_{j\in N} \lambda^e_{o'jo'o} = 1 \quad \forall e \in E, (o,o') \in T_e, (o',j) \in A \cup A' \tag{20}$$

$$\sum_{i\in N} \lambda^e_{ioo'o} = 1 \quad \forall e \in E, (o,o') \in T_e, (i,o) \in A \cup A' \tag{21}$$

$$\sum_{i\neq j, k\in N} \lambda^e_{ijo'o} = \sum_{j\neq k, k\in N} \lambda^e_{jko'o} \quad \forall e \in E, (o,o') \in T_e, j \in N \tag{22}$$

$$\lambda^e_{ijoo'} + \lambda^e_{ijo'o} \leq 1 \quad \forall e \in E, (o,o') \in T_e, (i,j) \in A \cup A' \tag{23}$$

$$\sum_{(i,j)\in A\cup A'} \lambda^e_{ijoo'} t^3_{ij} + \sum_{(i,j)\in A\cup A'} \lambda^e_{ijo'o} t^3_{ij} \leq H_e VOT_t \quad \forall e \in E, (o,o') \in T_e \tag{24}$$

$$\xi^e_{ikoo'} = \lambda^e_{ikoo'} + \sum_{j\in N} \xi^e_{ijoo'} \lambda^e_{jkoo'} \quad \forall e \in E, (o,o') \in T_e \tag{25}$$

$$\xi^e_{iko'o} = \lambda^e_{iko'o} + \sum_{j\in N} \xi^e_{ijo'o} \lambda^e_{jko'o} \quad \forall e \in E, (o,o') \in T_e \tag{26}$$

$$\prod_{e\in E, (o,o')\in T_e} (1-\xi^e_{pqoo'})(1-\xi^e_{pqo'o}) = 1 - w_{pq} \quad \forall (p,q) \in W \tag{27}$$

$$k_{ij},\ y_l \geq 0 \text{ are integers } \forall (i,j) \in A \cup A', l \in L \tag{28}$$

$$z_{ij}, \lambda^e_{ijoo'}, u_{l'} \in \{0,1\} \quad \forall (i,j) \in A \cup A',\ l' \in L',\ e \in E, (o,o') \in T_e \tag{29}$$

$$\begin{aligned} &\sum_{(i,j)\in A\cup A'} [(1-\mathit{\Lambda}_{ij})\ (t^1_{ij} + \rho_{ij})\ (x_{ij} - x^*_{ij}) + \mathit{\Lambda}_{ij}\ (t^2_{ij} + \rho_{ij})\ (x_{ij} - x^*_{ij})] + \\ &\sum_{(\hat{i},\hat{j})\in\hat{A}(\mathbf{y})} (\hat{t}_{\hat{i}\hat{j}} + \tilde{t}_{\hat{i}\hat{j}} + t^a_{\hat{i}\hat{j}} + \hat{\rho}_{\hat{i}\hat{j}})(\hat{x}_{\hat{i}\hat{j}} - \hat{x}^*_{\hat{i}\hat{j}}) - \sum_{(p,q)\in\hat{W}(\mathbf{y})} \left(\frac{1}{\theta} \ln \frac{\hat{d}^*_{pq}}{\bar{d}_{pq} - \hat{d}^*_{pq}} + \psi \right) . (\hat{d}_{pq} - \hat{d}^*_{pq}) \geq 0 \quad \forall x_{ij}, \hat{x}_{\hat{i}\hat{j}}, \hat{d}_{pq} \in \Omega(\mathbf{y}) \end{aligned} \tag{30}$$

$$\Omega\left(\mathbf{y}\right)=\left\{\begin{array}{l} x_{ij},\hat{x}_{\hat{i}\hat{j}},\hat{d}_{pq} \mid x_{ij} \le Mz_{ij} \quad \forall(i,j)\in A\cup A';\ X_r^{pq},\hat{X}_{\hat{r}}^{pq}\ge 0\ \ \forall r\in R_{pq},\hat{r}\in\hat{R}_{pq}\left(\mathbf{y}\right); \\ x_{ij}=\sum_{(p,q)\in W}\sum_{r\in R_{pq}} X_r^{pq}\delta_{ijr} \quad \forall(i,j)\in A\cup A';\ \hat{x}_{\hat{i}\hat{j}}=\sum_{(p,q)\in\hat{W}}\sum_{\hat{r}\in\hat{R}_{pq}(\mathbf{y})}\hat{X}_{\hat{r}}^{pq}\hat{\delta}_{\hat{i}\hat{j}\hat{r}} \quad \forall(\hat{i},\hat{j})\in\hat{A}\left(\mathbf{y}\right); \\ d_{pq}=\pi\sum_{r\in R_{pq}} X_r^{pq} \quad \forall(p,q)\in W;\ \hat{d}_{pq}=\sum_{\hat{r}\in\hat{R}_{pq}(\mathbf{y})}\hat{X}_{\hat{r}}^{pq} \quad \forall(p,q)\in\hat{W}\left(\mathbf{y}\right)=\left\{\ (p,q) \mid w_{pq}=1\ \right\}; \\ \bar{d}_{pq}=d_{pq}+\hat{d}_{pq} \quad \forall(p,q)\in W;\ \hat{d}_{pq}=0 \quad \forall(p,q)\in W\setminus\hat{W}\left(\mathbf{y}\right); \\ 0\le\hat{d}_{pq}\le\bar{d}_{pq} \quad \forall(p,q)\in\hat{W}\left(\mathbf{y}\right) \end{array}\right\} \qquad (31)$$

As shown above, this problem has four objectives. The first objective of our problem as stated in (1) is to maximize total user benefit. This objective is commonly considered in UTNDPs and seeks to benefit all users of the network on average regardless of the mode they use. On the other hand, encouraging travelers for more use of bus mode is a common approach taken by government authorities. Thus the second objective as depicted by (2) seeks to maximize the passenger demand share of the bus mode. The third objective as given in (3) is to maximize the total demand coverage provided by the bus network (i.e., the total demand for OD pairs connected by the bus network, divided by the total travel demand in the network.). The third objective aims at maximizing accessibility by covering as many demand nodes by bus network as possible (i.e., service coverage), and at the same time minimizing the disconnectivity of OD pairs within the bus network (i.e., maximizing the connectivity of the bus network). Note that this is not the same as the second objective, since demand share for some covered OD pairs may be low at the design time, but it can be planned to increase the share by improving frequencies etc. The last objective as shown in (4) is to minimize the average generalized travel cost between OD pairs connected through the bus network. In other words, it seeks to minimize the average generalized travel cost for a bus mode trip.

Constraints (5)-(14) are related to design constraints for the automobile network. Constraint (5) imposes a budget limit to the total cost of lane additions and link constructions. Constraint (6) limits the number of lanes added to the existing network links. Constraint (7) is the lane allocation equation for each link. Constraint (8) assures that at least one arc is present for each existing link. Constraint (9) is similar to (7) but is for candidate links. Constraints (10)-(11) ensure that at least one arc is built for each candidate link to be constructed. Constraints (12)-(13) assure that no lane is assigned to non-existing arcs. Constraint (14) requires that the minimum travel time cost between each pair of nodes is less than a very large value. This constraint ensures the strong connectivity of the network; i.e., it ensures that a route exists between every pair of nodes.

Constraints (15)-(27) are bus network design constraints. Constraint (15) ensures that bus routes are not built on non-existing arcs. Constraint (16) requires that at least one bus line passes through a bimodal arc and no bus lines passing through an auto arc. Constraints (17)-(19) and (20)-(22) respectively restrict the forth and back routes for bus lines to be built between the assigned terminal pairs. Constraints (17)-(18) and (20)-(21) assure that only one arc is assigned to the start arcs of forth and back routes of each bus line respectively. Constraints (19) and (22) assure that forth and back routes between terminal nodes pair for each bus line are connected and without cycles and backtracks. Constraint (23) restricts that the forth and back routes of each bus line do not pass the same arc to avoid the confusion of passengers. Constraint (24) restricts the round trip time of each bus line. In this paper, it is assumed that a feasible frequency has already been defined for each line after considering the available fleet and desired bus line frequencies (e.g., the frequencies of the basic unimproved transit network or new frequencies based on specific policies). The predefined frequencies and the maximum allowable round trip time implicitly impose transit budget limit on operational cost, which is directly proportional to the number of buses serving each line and roughly proportional to the travel distance or travel time of buses. Constraints (25)-(27) determine whether an OD pair is connected by the bus network.

Constraints (28)-(29) impose conditions to design variables' domains. Constraints (30)-(31) present the variational inequality formulations of the combined modal-split/assignment problem as the lower level problem, which determines the deterministic user equilibrium traffic flows and travel demands for the auto and bus modes. Since the auto and bus travel times are interdependent and the effect of one mode on other is not symmetric, it is not possible to formulate the problem as an equivalent minimization program (e.g., Dafermos, 1972; Abdulaal and LeBlanc, 1979; Sheffi, 1985). The problem can be stated as the finite-dimensional variational inequality (30) in terms of arc flows and bus demand, subject to constraint set (31). It is assumed that the bus passenger demand of an OD pair can only have a positive value when at least one bus route exists between that OD pair. i.e., only the travel demand between OD pairs connected by the bus network can use the bus mode. For other OD pairs, no bus passenger demand exists. In other words, OD demand for the bus mode can have a positive value only if the OD pair is connected by the bus network.

3.6. Modal Split and Travel Cost Functions

The travel demand share of auto and bus modes between each OD pair is calculated using the logit formula, which is widely used as a mode split function. The demand for the bus mode is obtained by:

$$\hat{d}_{pq} = \bar{d}_{pq} \frac{1}{1 + e^{\theta(h_{pq} - \hat{h}_{pq} - \psi)}} \tag{32}$$

where h_{pq} and $\hat{h}_{pq}$ are the minimum generalized travel costs of bus passengers and autos for OD pair (p, q). ψ and θ are the parameters of the logit modal split function. The minimum generalized travel costs are calculated as follows:

$$h_{pq} = \sum_{(i,j)\in A\cup A'} \sigma_{pqij}\left[(1-\Lambda_{ij})t^1_{ij} + \Lambda_{ij}t^2_{ij} + \rho_{ij}\right] \tag{33}$$

$$\hat{h}_{pq} = \sum_{(\hat{i},\hat{j})\in \hat{A}(\mathbf{y})} \hat{\sigma}_{pq\hat{i}\hat{j}}(\hat{t}_{\hat{i}\hat{j}} + \tilde{t}_{\hat{i}\hat{j}} + t^a_{\hat{i}\hat{j}} + \hat{\rho}_{\hat{i}\hat{j}}) \tag{34}$$

where σ_{pqij} and $\hat{\sigma}_{pq\hat{i}\hat{j}}$ are respectively equal to 1 if arcs (i, j) and $(\hat{i}, \hat{j})$ are on the shortest auto path and shortest bus hyperpath connecting OD pair (p, q) and are equal to zero otherwise. Since bus and auto flows are assumed to be interdependent to each other, different travel cost functions for the possible cases are formulated. The travel cost functions are as follows:

- The travel cost function for autos on auto arcs is defined by the well-known BPR function (see U.S. Bureau of Public Roads, 1964; Szeto and Lo, 2005; 2006; 2008; Lo and Szeto, 2009) and the value in-vehicle travel time VOT_t, and is given as follows:

  $$t^1_{ij} = t^0_{ij}\left(1 + \alpha\left(\frac{x_{ij}}{c_{ij}(k_{ij})}\right)^{\beta}\right)VOT_t \tag{35}$$

  where α and β are parameters, $c_{ij}(k_{ij})$ is the capacity of arc (i, j) dependent only on k_{ij}, and t^0_{ij} is the free flow travel time of automobiles on arc (i, j) in minutes.

- The travel cost function of autos on bimodal arcs is defined by equation (34) (see Uchida *et al.*, 2007):

  $$t^2_{ij} = t^0_{ij}\left(1 + \alpha\left(\frac{\mu\bar{f}_{ij} + x_{ij}}{c_{ij}(k_{ij})}\right)^{\beta}\right)VOT_t \tag{36}$$

  where parameter μ converts the bus flow into the equivalent auto flow, $\bar{f}_{ij}$ is the total frequency of buses on arc (i, j). i.e., the fixed flow of buses passing through arc (i, j), which is computed by $\bar{f}_{ij} = \sum_{e\in E} f_e\left(\lambda^e_{ijoo'} + \lambda^e_{ijo'o}\right)$, where f_e is the frequency of bus line e per hour; $\lambda^e{}_{ij}$ is the binary variable, which equals 1 if bus line e passes through arc (i, j) and equals zero otherwise.

- The travel cost function of bus passengers on bimodal arcs is similar to that of auto in bimodal arcs and is defined as follows:

  $$t^3_{ij} = \hat{t}^0_{ij}\left(1 + \alpha\left(\frac{\mu\,\bar{f}_{ij} + x_{ij}}{c_{ij}(k_{ij})}\right)^{\beta}\right)VOT_t \tag{37}$$

  Specific cost functions are required for the arcs of the modified bus network. Each arc is associated with an in-vehicle travel time cost, an access time cost, a waiting time cost, and a bus fare. The in-vehicle travel cost is calculated based on the in-vehicle travel time function proposed by Uchida *et al.* (2007) and the value of in-vehicle travel time:

$$\tau^{e}_{\hat{i}\hat{j}} = \left[\sum_{(i,j)\in \tilde{A}_{\hat{i}\hat{j}}(\mathbf{y})} t^{3}_{ij} \right] \left[1 + \alpha' \left(\frac{\chi^{e}_{\hat{i}\hat{j}} + \tilde{\chi}^{e}_{\hat{i}\hat{j}}}{f_e n_e} \right)^{\beta'} \right] \tag{38}$$

where α' and β' are the parameters of the in-vehicle congestion effect function, and n_e is the passenger capacity of the buses of line *e*. The first square bracket term in (38) is in-vehicle travel time cost on arc $(\hat{i}, \hat{j})$, whereas the second square bracket term is a BPR type function modeling in-vehicle congestion, which considers the total number of passengers who compete for line *e*. The line passenger flow on arc $(\hat{i}, \hat{j})$ is obtained using the following relation (De Cea and Fernández, 1993):

$$\chi^{e}_{\hat{i}\hat{j}} = \hat{x}_{\hat{i}\hat{j}} \frac{f_e}{\sum_{e\in \bar{E}_{\hat{i}\hat{j}}(\mathbf{y})} f_e} \tag{39}$$

Having the in-vehicle travel time cost of bus lines *e* on arc $(\hat{i}, \hat{j})$, the average value of in-vehicle travel time costs for all bus lines passing the arc can be obtained by the following relation (Uchida *et al.*, 2007):

$$\hat{t}_{\hat{i}\hat{j}} = \frac{\sum_{e\in \bar{E}_{\hat{i}\hat{j}}(\mathbf{y})} \tau^{e}_{\hat{i}\hat{j}} f_e}{\sum_{e\in \bar{E}_{\hat{i}\hat{j}}(\mathbf{y})} f_e} \tag{40}$$

The time cost required to access the bus station on arc $(\hat{i}, \hat{j})$ via walking is defined as follows:

$$t^{a}_{\hat{i}\hat{j}} = \bar{t}_{\hat{i}\hat{j}}.VOT_a \tag{41}$$

where VOT_a is the value of walking time, and $\bar{t}_{\hat{i}\hat{j}}$ is the average walking time to reach the bus station on arc $(\hat{i}, \hat{j})$ in minutes.

Finally, the waiting time cost associated with arc $(\hat{i}, \hat{j})$ is defined as:

$$\tilde{t}_{\hat{i}\hat{j}} = \frac{60}{\sum_{e\in \bar{E}_{\hat{i}\hat{j}}(\mathbf{y})} f_e} VOT_w + \alpha' \left(\frac{\hat{x}_{\hat{i}\hat{j}} + \tilde{x}_{\hat{i}\hat{j}}}{\sum_{e\in \bar{E}_{\hat{i}\hat{j}}(\mathbf{y})} f_e n_e} \right)^{\beta'} VOT_w \tag{42}$$

where α' and β' are the parameters of the function, and VOT_w is the value of (extra) waiting time. The first term denotes the expected waiting time cost, assuming a uniform arrival distribution for passengers and an exponential headway distribution for buses as in Uchida *et al.* (2007). The last term is the extra waiting time cost due to the insufficient capacity of bus line passing through arc $(\hat{i}, \hat{j})$. i.e. in-vehicle congestion (De Cea and Fernández, 1993; Gao *et al.*, 2004; Uchida *et al.*, 2007; Szeto *et al.*, 2011). Insufficient bus capacity may prevent the passengers to board the first arriving bus and thus lead to increase their individual waiting times. This extra waiting time is modeled by the power function $\alpha'\left(\frac{\hat{x}_{\hat{i}\hat{j}}+\tilde{x}_{\hat{i}\hat{j}}}{\sum_{e\in\bar{E}_{\hat{i}\hat{j}}(\mathbf{y})} f_e n_e}\right)^{\beta'}$. The set of attractive lines $\bar{E}_{\hat{i}\hat{j}}(\mathbf{y})$ for each arc $(\hat{i},\hat{j})$ is calculated using the following mathematical model (De Cea and Fernández, 1993):

$$\min \quad \frac{1+\sum_{e\in\hat{E}_{\hat{i}\hat{j}}(\mathbf{y})} \tau_{\hat{i}\hat{j}}^{e} f_e \upsilon_e^{\hat{i}\hat{j}}}{\sum_{e\in\hat{E}_{\hat{i}\hat{j}}(\mathbf{y})} f_e \upsilon_e^{\hat{i}\hat{j}}} \tag{43}$$

subject to

$$\upsilon_e^{\hat{i}\hat{j}} \in \{1,0\}, \ \sum_{e\in\hat{E}_{\hat{i}\hat{j}}(\mathrm{y})} \upsilon_e^{\hat{i}\hat{j}} \geq 1 \tag{44}$$

where $\upsilon_e^{\hat{i}\hat{j}}$ is the binary variable which equals 1 if bus line *e* is an attractive line on arc $(\hat{i},\hat{j})$, and equals 0 otherwise.

4. Solution Procedures for the Problem

Previous studies indicate that bi-level programming problems are NP-hard. As discussed by Ben-Ayed *et al.* (1988), even a simple linear bi-level programming problem is still NP-hard. Moreover, many bi-level programming problems are non-convex. There is no guarantee for the whole problem to be convex even when both upper and lower level problems are convex. Furthermore, although Branch and Bound algorithms can be applied to solve these problems optimally in small sizes, they are not able to solve real size problems. Consequently, in recent years, there has been an increasing interest in solving urban transportation network design problems using metaheuristic algorithms and especially their hybrid versions. In the previous work of the authors (Miandoabchi *et al.*, 2010; 2011), hybrid multi-objective metaheuristics are proposed to solve the proposed problem; a hybrid of genetic algorithm and simulated annealing, a hybrid of particle swarm optimization and simulated annealing, and a hybrid of harmony search algorithm and simulated annealing. A distance-based fitness value calculation method is used in these algorithms.

In the present paper, two hybrid multi-objective population-based metaheuristic algorithms are proposed to solve the problem: A multi-objective Hybrid Genetic Algorithm (mHGA), and a new multi-objective Hybrid Clonal Selection Algorithm (mHCSA) which is a variation of immune based methods, hybridized with simulated annealing. mHCSA and even its single form without hybridization have never been used to solve urban transportation network design problems before.

The fitness value calculation method is adapted from SPEA2 (Strength Pareto Evolutionary Algorithm2), a multi-objective genetic algorithm proposed by Zitzler *et al.* (2001). In the next sub-section, the common attributes and general characteristics of the algorithms are described.

4.1. Similarities and Comparison of the Proposed Algorithms

In this sub-section the common characteristics of the developed algorithms are described.

Pareto-optimal Set Generation

Since the objective functions are not combined using weights, the algorithms deal with separate objective functions rather than a single objective function. This requires defining a set of Pareto-optimal solutions to maintain the list of non-dominated solutions throughout the solution. A solution is added to the set after being checked for non-dominance by other solutions and the dominated solutions are omitted from the set if necessary. The fitness calculation method used in this paper requires that the Pareto-optimal set is updated at the end of each iteration in the developed algorithms, instead of checking the set for updating every time when a new solution is generated.

Calculation of Fitness Values

As noted earlier, the fitness value calculation method used here is adapted from SPEA2 algorithm and is changed slightly to fit the problem under study. In order to compute the fitness function of a new solution, a value S needs to be defined for each solution O in the Pareto-optimal set or population. The calculation or recalculation of S values is under taken after the generation of initial population, and at the end of each iteration of the algorithms, when the population and Pareto-optimal set are updated. To do this, first the population P and Pareto-optimal set $\overline{P}$ are combined to form a union set, and then for each solution O the number of dominating solutions in the union set are found. Then the value of $S(O)$ is obtained using (45).

$$S(O)=\frac{\left|\left\{a \mid a\in P\cup\overline{P}, a\succ O\right\}\right|}{\left|P\cup\overline{P}\right|} \tag{45}$$

The fitness value of a solution O' is composed of two values; one is related to the number of solutions that dominate the solution, and the other is a density estimate value which is obtained using the k^{th} nearest neighbor method. The lower the value of the fitness, the better is the solution. The fitness calculation procedure is described below:

- Find the solutions in both the population and Pareto-optimal set which dominate O', and calculate the first part of the fitness value:

$$R(O') = \sum_{O \in P \cup \overline{P}, O \succ O'} S(O) \tag{46}$$

- Compute the relative distance of O' to each solution O in the Pareto-optimal set or population, where $Z_i^{\min}$ is the minimum value of the minimization objective Z_i, $Z_j^{\max}$ is the maximum value of the maximization objective function Z_j:

$$d(O', O) = \sqrt{\sum_{i=2,3,4} \left(\frac{Z_i(O) - Z_i(O')}{Z_i^{\min}} \right)^2 + \sum_{j=1} \left(\frac{Z_j(O) - Z_j(O')}{Z_j^{\max}} \right)^2} \tag{47}$$

- Sort the values of $d(O',O)$ in an ascending order, and select the kth lowest value of $d(O,O')$ as the kth nearest neighbor to solution O', σ^k where $k = \sqrt{|P \cup \overline{P}|}$.

- Calculate the density estimate as the second part of the fitness value:

$$D(O') = \frac{1}{\sigma^k + 2} \tag{48}$$

- Calculate the fitness value of solution O' as follows:

$$F(O') = R(O') + D(O') \tag{49}$$

The fitness value calculation method used in this paper slightly differs from its original version. A denominator is added in (45) to normalize the S values because of the large variations in the values since they depend on the number of Pareto-optimal and population solutions. Also the objective values are normalized in the distance calculation step because of the significant differences between value ranges of the four objective functions.

The values of the objective functions Z_1, Z_2, Z_3, and Z_4 in (47) are respectively calculated after having the minimum generalized travel cost by mode by OD pair, the demand for the bus mode for each OD pair, the list of OD pairs connected by the bus network, and the minimum travel cost of bus passengers between each OD pair, which are obtained by solving the lower level combined modal-split/assignment problem (30)-(31) using the following procedure:

1. Set all the $\upsilon_e^{\hat{i}\hat{j}}$ values to 1.

2. Based on the values of $\upsilon_e^{\hat{i}\hat{j}}$**,** solve VI (30)-(31) to obtain $\tau *_{\hat{i}\hat{j}}^{e}$, i.e., the optimal value for $\tau_{\hat{i}\hat{j}}^{e}$.

3. Based on the values of $\tau *_{\hat{i}\hat{j}}^{e}$, solve problem (43)-(44) to find $\upsilon *_e^{\hat{i}\hat{j}}$.

4. If $\upsilon *_e^{\hat{i}\hat{j}} = \upsilon_e^{\hat{i}\hat{j}}$, stop. Otherwise, set $\upsilon_e^{\hat{i}\hat{j}} = \upsilon *_e^{\hat{i}\hat{j}}$. Go to step 2.

In step 2, the VI is solved using the “diagonalization algorithm”. It consists of iteratively solving a series of equilibrium sub-problems. At each iteration of the diagonalization algorithm, the sub-problem with the following objective function to be solved subject to constraint set (31) (see Sheffi, 1985):

$$\text{Min } z = \sum_{(i,j)\in A\cup A'} \int_0^{x_{ij}} \left[(1-\Lambda_{ij})t_{ij}^{1}(w) + \Lambda_{ij} t_{ij}^{2}(w) + \rho_{ij} \right] dw + \\ \sum_{(\hat{i},\hat{j})\in \hat{A}(\mathbf{y})} \int_0^{\hat{x}_{\hat{i}\hat{j}}} \left[\hat{t}_{\hat{i}\hat{j}}(w) + \tilde{t}_{\hat{i}\hat{j}}(w) + t_{\hat{i}\hat{j}}^{a} + \hat{\rho}_{\hat{i}\hat{j}} \right] dw + \sum_{(p,q)\in \hat{W}(\mathbf{y})} \int_0^{\hat{d}_{pq}} \left(\frac{1}{\theta} \ln \frac{w}{\bar{d}_{pq} - w} + \psi \right) dw \tag{50}$$

Dafermos (1982) and Florian and Spiess (1982) have discussed the sufficient conditions for the convergence of the diagonalization algorithm for asymmetric traffic equilibrium assignment problems and multimodal traffic assignment problems including ours. The former gives the global convergence results under the assumption of the uniformly monotone link cost function. The latter gives a sufficient condition for the local convergence under the assumption of monotone link cost function and $\left\| \mathbf{I} - \mathbf{D}(\mathbf{v}^*)^{-1/2} \mathbf{B}(\mathbf{v}^*) \mathbf{D}(\mathbf{v}^*)^{-1/2} \right\| < 1$, where $\mathbf{B}(\mathbf{v}^*)$ is the Jacobian of the link cost function at equilibrium link flows $\mathbf{v}^*$ and $\mathbf{D}(\mathbf{v}^*) > \mathbf{0}$ is the diagonal of $\mathbf{B}(\mathbf{v}^*)$.

This diagonalization algorithm looks very crude, but it is commonly used to solve asymmetric traffic equilibrium and multimodal traffic equilibrium problems because of easy implementation (De Cea and Fernández, 1993). Moreover, as stated in De Cea and Fernández (1993), those conditions are sufficient for convergence but not necessary, and the diagonalization algorithm has shown to have good convergence properties in practice even when the monotonicity condition is not satisfied. In our problems, the convergence was observed in all our test instances.

Checking the Feasibility of Solutions

Each new solution generated by each algorithm is checked for feasibility. There are two types of feasibility checking, namely checking for the strong connectivity of the road network and checking for the route trip time of each bus line. For checking the strong connectivity, each solution generated by in each of the algorithms is checked in two phases. First, the algorithm checks whether the solution satisfies the "necessary" condition of the strong connectivity (i.e., all nodes have at least one outgoing and one incoming lane). If the solution satisfies the necessary condition, then the algorithm attempts to find the shortest paths between all OD pairs using Dijkstra's algorithm. If at least one of the OD pairs is disconnected, i.e., no shortest path can be found between this OD pair, then the generated network is rejected because it is impossible for this network to have one path connecting each OD pair. Because the first phase screens out some of infeasible solutions, this two phase checking procedure can help to reduce the use of the shortest path algorithm and hence computation time compared with just using the shortest path algorithm for checking.

Checking the feasibility of the round trip time of each bus line is performed after a bus network is generated from scratch or after a solution is modified by the SA. A solution is discarded if the round trip time of at least one bus line is found to be longer than the maximum allowable round trip time. Connectivity of bus routes is another important feasibility issue. However, this is not required to be checked in both algorithms, as the route construction heuristic and the bus network mutation sub-routine depicted later implicitly avoid the generation of any disconnected bus route.

Initial Population Generation

Each initial population solution is generated randomly by using the initial solution generation procedure. The procedure consists of three phases as shown in Figure 1.

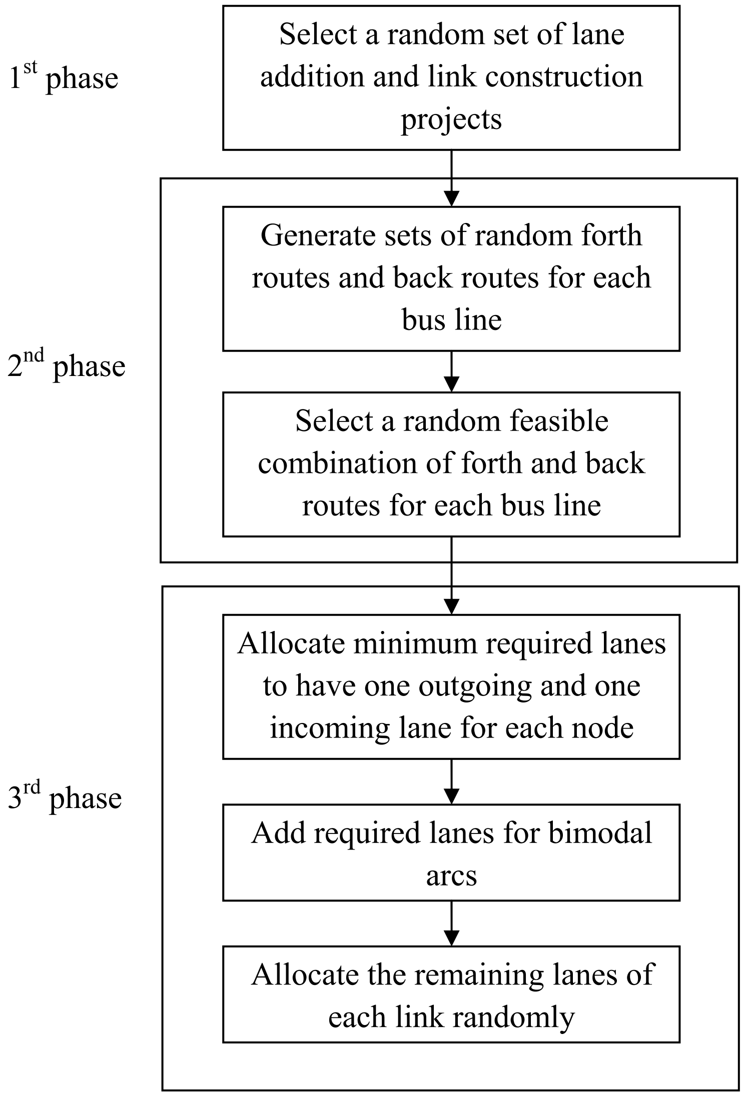
1st phase
Select a random set of lane addition and link construction projects
2nd phase
Generate sets of random forth routes and back routes for each bus line
Select a random feasible combination of forth and back routes for each bus line
3rd phase
Allocate minimum required lanes to have one outgoing and one incoming lane for each node
Add required lanes for bimodal arcs
Allocate the remaining lanes of each link randomly

Figure 1. Initial solution generation procedure

In the first phase, a random feasible set of lane addition or link construction projects is defined. In the second phase, forth and back routes for bus lines are defined by using a route construction heuristic procedure. First, a set of random forth routes and a set of random back routes of a bus line are constructed. Then, all possible combinations of forth and back routes are checked, and among the combinations with feasible round trip time of the bus line, one is selected randomly. The procedure is repeated for each bus line separately.

In order to generate sets of random forth and back routes for each bus line in the first part of the second phase, a heuristic sub-routine is developed. The idea of generating random route sets is to start from a shortest route between a pair of terminal nodes and to extend the route in a random manner until several random feasible routes are generated. Experiments indicate that this method can provide adequate diversity for bus routes. Applying the shortest path concept offers two advantages; first, connected routes can be generated and modified easily without the need of time consuming path enumeration methods, and second, the routes are generated and modified in a controlled manner so that the resulting routes may not have unacceptably long round trip times.

To do this, the following procedure is carried out once for generating a set of forth routes of a bus line and once for generating a set of back routes of that bus line:

- Find the shortest path between two terminals of the bus line and add the path to the set of routes.
- Repeat the following until the travel time of the last generated route exceeds the maximum allowable round trip time of the bus line:
    - Select the last generated route in the route set, and choose a random sub-sequence of its nodes.
    - Generate the shortest route connecting the start and end nodes of the sub-sequence without passing any arc on the last generated route (to avoid generation of cycles and backtracks).
    - Replace the selected sub-sequence in the last generated route with the generated shortest route to obtain a new route.
    - Add the new route to the set of routes.

Note that free flow bus travel times are used for the shortest path calculations, since the equilibrium travel times are not known at this stage.

Finally, in the third phase, lanes are allocated to network arcs. The phase consists of three steps. In order to reduce the possibility of generating disconnected networks, in the first step, lanes are allocated in such a way that each node has at least one outgoing and one incoming lane. This can ensure that the necessary condition for network connectivity is satisfied. To do this, first, for each link, a lane is allocated randomly to one of the arcs to have at least one lane on each link. Next, for each node without outgoing lanes, one lane is allocated to one of its outgoing arcs, and for each node without incoming lanes, one lane is allocated to one of its incoming arcs. In the second step, the connectivity of bus line routes is checked. If there is no lane on a bimodal arc but at least a bus route must pass through this arc, a lane is allocated to this arc. The last step is to randomly allocate the remaining lanes to the network arcs.

At the end, the generated solution is discarded if the network is disconnected and the three phases are repeated again to obtain another initial solution.

Comparison of the Proposed Algorithms

Before proceeding to the detailed description of the developed algorithms, their solution generation methods and evolution strategies are compared and depicted in Table 1.

Table 1. Comparison of the developed algorithms

Algorithm	Number of iterations	Solution generation method	Evolution strategy
mHGA	*G* generations	Selecting two parents, applying crossover, and applying SA on some of the offspring solutions	Replacing a number of population solutions with offspring solutions
mHCSA	*T* iterations	Selecting a number of solutions, making copies of them, and applying SA on the produced copies.	Replacing each selected population solution with the copied and improved solutions

4.2. Multi-objective Hybrid Genetic Algorithm

Genetic algorithm (GA) is an evolutionary algorithm that imitates the genetic evolution process of the living species. GA is one of the most well-known population-based solution methods, which was first proposed by Holland (1975). It has been used to solve various network design problems (e.g., Drezner and Salhi, 2002; Drezner and Wesolowsky, 2003; Cantarella *et al.*, 2006; Cantarella and Vitetta, 2006; Long *et al.*, 2010). Miandoabchi *et al.* (2010; 2011) proposed two hybrid versions of GA and simulated annealing (SA) as one of the solution algorithms for their problems. The computational results in these previous studies suggested that the hybrid GAs outperformed other hybrid solution algorithms with SA such as a hybrid particle swarm optimization, a hybrid harmony search, a hybrid artificial bee colony algorithm, and a hybrid evolutionary algorithm. Following the previous works, here another hybrid GA algorithm is developed to solve the problem, along with another population based solution method, mHCSA. Although the GA proposed here is hybridized with SA, it has similarities with the previous algorithms only in some parts of the initial solution generation phase and in the evolution mechanism. The algorithm used here differs from the two versions in four aspects. First, the proposed algorithm applies a multi-phase crossover operator which is specifically designed for this problem. Second, the proposed algorithm uses a different fitness value calculation method. Third, the proposed algorithm includes additional considerations related to the bus mode decisions. Fourth, the proposed algorithm uses a different move generation method in SA. General steps of the algorithm are presented below and the details are described later.

Phase 1: Generate an initial population of *NP* solutions:

- Repeat the following until *NP* solutions are generated:
    - Generate an initial solution randomly using the initial solution generation procedure.
    - Check the road network connectivity, and the feasibility of the round trip time of each bus line.
    - If the solution is feasible, add it to the population; otherwise discard it.
- Compute the *S* values for population solutions.
- Build the initial Pareto-optimal set from the initial population.
- Compute the fitness values of population and Pareto-optimal solutions.

Phase 2: Repeat the following procedure for *G* generations:

- Repeat for an offspring set with *NO* feasible offspring solution is generated:
    - Select the first parent randomly from population (using the roulette wheel selection rule) or Pareto-optimal solutions (using pure random selection).
    - Select the second parent randomly from population (using the roulette wheel selection rule) or Pareto-optimal solutions (using pure random selection).
    - Apply the crossover operator to obtain two offspring solutions.
    - Check the feasibility of each offspring in terms of network connectivity and discard it if the network is disconnected.
    - Check the budget feasibility of each offspring and apply the budget reduction sub-routine if needed. If the infeasible solution cannot be repaired, then discard the offspring.
    - Compute the fitness value for each feasible offspring solution.
- Replace the population solutions using the evolution mechanism:
    - Select the population solutions to be eliminated and the offspring solutions to be inserted into the population.
    - Apply SA on a proportion of the best selected offspring solutions and update the population.
- Update the Pareto-optimal set using all offspring and the solutions generated by SA.
- Recalculate *S* values and fitness values of population and Pareto-optimal set.

Solution Encoding

Each solution is encoded as a chromosome. The chromosome contains the information such as the street directions, lane allocations, lane or link additions, and finally bus routes. Due to the variety of solution attributes, two chromosome presentations are used. Both configurations of the road network and the bus network are depicted by a 2-rows matrix chromosome. In the road network matrix chromosome, each column corresponds to an existing or a candidate link and thus the total length of each chromosome equals $L+L'$. The first and second rows of each column denote the lane allocations of arcs (i, j) and (j, i) of link l or l'. The bus network matrix chromosome has two rows, each indicating the forth and back routes of the bus lines in the form of a sequence of road network nodes. The complete chromosome is the concatenation of routes of all bus lines. The length of the chromosome is not fixed and depends on the number of nodes covered by bus lines.

Figure 2 depicts a typical 8-node 11-link network, and figure 3 illustrates its chromosome presentations. The two dashed and dotted lines between nodes 3-7, and 7-5 are the candidate links, which are not included in the current network design scenario. Therefore, the chromosomoe representation have 13 columns. One-head bold arrows represents one-way streets and two-head bold arrows represents two-way streets. Dashed arrows form the bus network skeleton. Two bus lines operate on the network: bus line 1 which has arcs 1-2, 2-5, and 5-8 as its forth route and on arcs 8-5, 5-4, 4-2, 2-1 as its back route, and bus line 2 which has arcs 1-3, 3-6, 6-7, 7-4, and 4-5 as its forth route and arcs 5-4, 4-7, 7-6, 6-3, 3-1 as its back route.

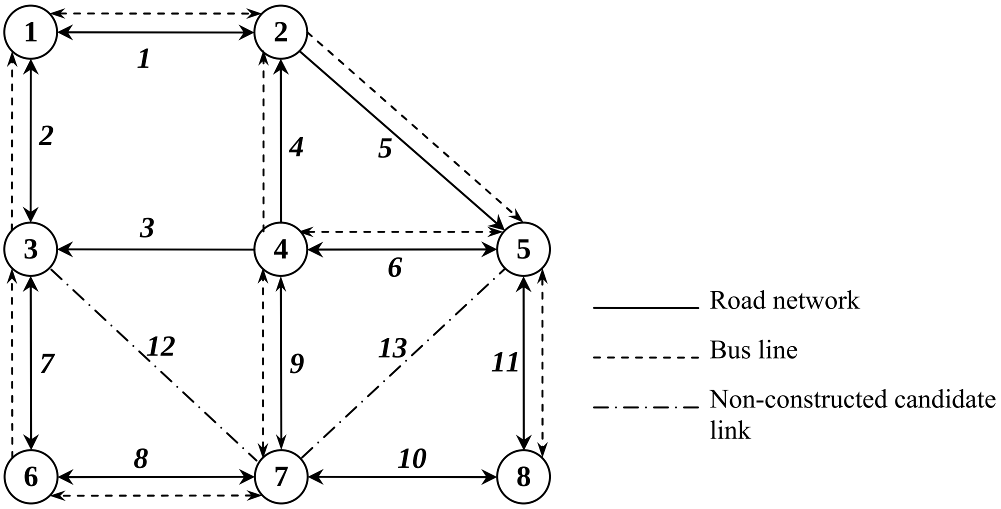

Figure 2. A typical network

	1	2	3	4	5	6	7	8	9	10	11	12	13
k_{ij}	2	3	0	0	2	1	3	2	2	2	2	0	0
k_{ji}	2	1	2	2	2	1	1	2	2	2	0	0	0

Figure 3(a). Road network chromosome

Forth	1	2	5	8	-	1	3	6	7	4	5
Back	8	5	4	2	1	5	4	7	6	3	1

Figure 3(b). Bus network chromosome

Crossover Operator

The crossover operator used here is a specific operator customized for the problem under study. The operator combines two parents to form two offspring in three phases.

In the first phase, a uniform crossover is used for generating the bus network chromosome of an offspring. That is, forth and back routes for each bus line are selected randomly from one of the parents to define the bus network chromosome of the first offspring. A second offspring is obtained by reversing all the selections. Figure 4 depicts a typical application of the proposed crossover operator on the chromosomes of three bus lines.

Forth	1	12	8	2	-	4	5	6	7	8	2	1	5	9	13	-
Back	2	10	7	12	1	2	8	7	6	5	4	13	10	6	12	1

Parent 1

Forth	1	5	6	7	8	2	4	5	9	10	11	2	1	5	6	10	13
Back	2	11	10	6	12	1	2	11	10	9	4	-	13	10	6	12	1

Parent 2

⬇

Forth	1	12	8	2	-	4	5	9	10	11	2	1	5	6	10	13
Back	2	10	7	12	1	2	11	10	9	4	-	13	10	6	12	1

Offspring 1

Forth	1	5	6	7	8	2	4	5	6	7	8	2	1	5	9	13	-
Back	2	11	10	6	12	1	2	8	7	6	5	4	13	10	6	12	1

Offspring 2

Figure 4. Typical application of the crossover operator

In the second phase, the lane allocation for each link in the road network chromosome of the offspring is determined by one of the parents. The lane allocations of links with auto arcs in both directions are randomly determined by one of the parents. The lane allocations for bimodal links are slightly more complicated. For a link with a bimodal arc for at least one direction as depicted by the offspring, there are four possible states for each of the two arcs of that link as shown in Figure 5, depending on the source of the bus lines on that arc. State A (B) refers to the case that the source is from parent 1 (2) only. State C refers to the case that the source is from both parents 1 and 2. State

D refers to the case that there is no bus line on the arc. Depending on the combination of the states of the pair of arcs on the bimodal link considered, the lane allocation to that link is determined based on the following set of rules.

- One arc has state A, and the other arc has state A or C or D: the lane allocation is determined by the first parent.
- One arc has state B, and the other arc has state B or C or D: the lane allocation is determined by the second parent.
- One arc has state C, and the other arc has state C or D: the lane allocation is randomly determined by one of the parents.
- One arc has state A, and the other arc has state B: the lane allocation is randomly determined by one of the parents.

The last selection rule may lead to lane allocations with zero lanes for one of bimodal arcs. Such infeasible solutions are repaired; if there is any bimodal arc with zero lanes, one lane is moved from the other arc on the same link to the current arc.

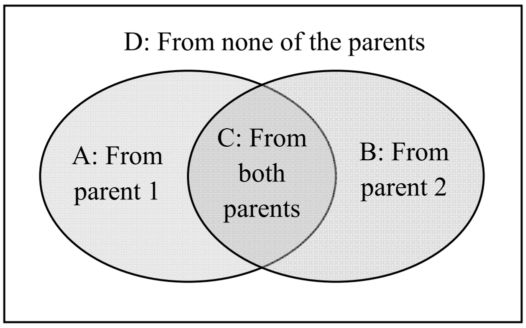

Figure 5. Possible states of bus lines passing through a bimodal arc

In the third phase, the obtained offspring is examined for feasibility. First, network connectivity is checked. If an offspring produces a disconnected road network, the offspring is discarded. Otherwise, the offspring is checked for budget feasibility. If the total construction cost is larger than the budget limit, a budget reduction sub-routine modified from Miandoabchi *et al*. (2011) is applied to try to repair the infeasible solution:

- Check every possible swap of lane addition or link construction projects; i.e., omission of an existing project and inclusion of another project.
- Exclude the swaps with candidate links with bimodal arcs.
- If there is at least one feasible swap after considering the budget limit and the network connectivity, select a random swap and apply it.
- If there is not a feasible swap, then check whether it is possible to omit at least one current project in terms of network connectivity.

- If there is at least one feasible omission, then repeat the following steps until the budget level is not exceeded or the network becomes disconnected:
    - Omit the project with maximum cost.
    - If the omitted project belongs to an existing link, reallocate the lanes to maintain the previous arc directions and in case of links with bimodal arcs to maintain the bus route connectivity too.
    - If the road network becomes disconnected, then report infeasibility and stop.

The key modification of the budget reduction subroutine of Miandoabchi *et al*. (2011) is to add a step to exclude the swaps including candidate links with bimodal arcs.

Embedded SA algorithm

Here, instead of using a classical mutation operator, a short version of SA is applied to a subset of offspring solutions as in Miandoabchi *et al.* (2010; 2011). However, the SA used in this paper has additional attributes compared to those in the previous works. SA as a mutation operator has two advantages compared to simple small perturbation mutation as in classic GA. First, SA can explore the neighborhood of the produced offspring in the hope of finding better solutions in the unexplored solution space areas. Second, SA can provide a series of random perturbations to the selected offspring solution. Nevertheless, because of the high computational requirements, SA is only applied on a subset of offspring solutions instead of all of them. In the developed SA, one iteration is performed per temperature. A move that generates a new solution consists of two sub-routines: One applies random perturbation to the bus network and the other applies random changes to the lane allocations.

The bus network mutation sub-routine applies random changes to each bus line; for each forth and each back route of each bus line, it is randomly decided to change the sequence of that route. The change in the sequence is performed by selecting a random sub-sequence of the route and by replacing it with a new sub-sequence. The new sub-sequence is generated in the same way explained in the second phase of the initial population generation procedure. Similarly, free flow travel times of arcs are used to generate shortest paths. Note that in some cases, no path may be found to substitute the selected sub-sequence. In such cases another sub-sequence is selected. Necessary changes in lane allocations are done after the replacement of the new sub-sequence to repair any disconnectivity in bus routes by allocating one lane to bimodal arcs with zero lanes. After applying random changes to each forth or back route of a bus line, the round trip time of the bus line is checked and if it exceeds the maximum allowable round trip time, the replacement of sub-sequence is discarded and another sub-sequence is selected randomly. The procedure is repeated until a feasible bus line in terms of round trip time is obtained.

The lane allocation mutation sub-routine applies changes to the lane allocations by iteratively selecting links and changing their lane allocations randomly; i.e., the links are selected one-by-one and the following steps are carried out for each chosen link: (1) Randomly decide to change the link, (2) Randomly change the lane allocation of the link.

Changing the allocation of lanes may lead to a disconnected network in two ways; first, the nodes of a link may have zero outgoing or incoming lanes; second, some node pairs may become disconnected, even when the first case does not occur. The latter is not easily predictable, but the former can be easily avoided by using specific feasibility constraints. The feasibility constraints can be used to find possible alternatives for lane allocations that do not lead the former case of disconnectivity. Moreover, the connectivity of bus line routes can be assured by using the feasibility constraint set. The set of general feasibility constraints for each link *l* or *l'* between nodes *i* and *j* is as follows:

$$k_{1ij} + k_{2ij} + \sum_{p \in \Phi_i,\, p \neq j} k'_{ip} \geq 1 \tag{51}$$

$$k_{1ji} + k_{2ji} + \sum_{p \in \Phi_i,\, p \neq j} k'_{pi} \geq 1 \tag{52}$$

$$k_{1ij} + k_{2ij} + \sum_{p \in \Phi_j,\, p \neq i} k'_{pj} \geq 1 \tag{53}$$

$$k_{1ji} + k_{2ji} + \sum_{p \in \Phi_j,\, p \neq i} k'_{jp} \geq 1 \tag{54}$$

$$k_{1ij} + k_{2ij} + k_{1ji} + k_{2ji} = k'_{ij} + k'_{ji} \tag{55}$$

$$k_{1ij} = \Lambda'_{ij} \tag{56}$$

$$k_{1ji} = \Lambda'_{ji} \tag{57}$$

k'_{ip}, k'_{pi}, k'_{jp}, and k'_{pi} are the current values for the lane allocations of the road links that are connected either to *i* or *j*. k'_{ij} and k'_{ji} denote the current lane allocation of the selected link (*i*, *j*) (i.e., for arcs (*i*, *j*) and (*j*, *i*), respectively). Λ'_{ij} and Λ'_{ji} indicate the presence or absence of bus routes on arcs of the selected link. Φ_i and Φ_j are correspondingly the sets of adjacent nodes to nodes *i* and *j*. The variables k_{1ij} and k_{1ji} indicate the minimum allowable number of lanes on arc (*i*, *j*) and (*j*, *i*) respectively and the variables k_{2ij} and k_{2ji} indicate the remaining number of lanes on those arcs. The above constraint set limits the possible alternatives for the lane allocation of a selected link while keeping current lane allocations of the other related links fixed such that the number of incoming and outgoing lanes of *i* and *j* nodes do not become zero. Moreover, they maintain the connectivity of bus line routes and ensure that at least one lane must be allocated to the arc of the selected link if required.

The second terms in (51) and (54) respectively calculate the total numbers of outgoing lanes from nodes *i* and *j* except the lanes of arcs (*i*, *j*) and (*j*, *i*), respectively. Similarly, the second terms in (52) and (54) calculate the total numbers of incoming lanes to nodes *i* and *j* except the lanes of arcs (*i*, *j*) and (*j*, *i*), respectively. The inequality pair (51)-(52) ensures that at least one outgoing and one incoming lane remain for node *i* whereas the inequality pair (53)-(54) ensure that at least one outgoing and one incoming lane remain for node *j*. Equation (55) is the link capacity allocation constraint. Finally, equations (56)-(57) imply that if arc (*i*, *j*) or (*j*, *i*) is bimodal, its minimum number of allowable lanes is 1 to assure the connectivity of bus line routes, and is zero otherwise. The values of k_{1ij} and k_{1ji} can be easily set using constraints (56)-(57). The remaining reduced set of constraints can be used to define the feasibility interval for k_{2ij}. The lower bound LB_{ij} and the upper bound UB_{ij} of the interval will be as follows:

$$LB_{ij} = \max\left\{0, \max\left\{1 - \sum_{p\in\Phi_j, p\neq i} k'_{pj} - \Lambda'_{ij}, 1 - \sum_{p\in\Phi_i, p\neq j} k'_{ip} - \Lambda'_{ij}\right\}\right\} \tag{58}$$

$$UB_{ij} = \min\left\{k'_{ij} + k'_{ji} - \Lambda'_{ij} - \Lambda'_{ji}, \min\left\{\begin{matrix} k'_{ij} + k'_{ji} - \Lambda'_{ij} + \sum\limits_{p\in\Phi_j, p\neq i} k'_{jp} - 1, \\ k'_{ij} + k'_{ji} - \Lambda'_{ij} + \sum\limits_{p\in\Phi_i, p\neq j} k'_{pi} - 1 \end{matrix}\right\}\right\} \tag{59}$$

It suffices to choose a random value for k_{2ij} from the interval, and then to obtain the value of k_{2ji} using constraint (55).

When a new solution is generated, it is checked for the connectivity of the road network, and the feasibility of the round trip time of each bus line. If the solution is infeasible, it is discarded and another solution is generated. This is repeated until the obtained solution is feasible.

Evolution Mechanism

The evolution process is performed by replacing some of the population solutions with the generated offspring solutions with higher fitness values as in Miandoabchi *et al.* (2011). The process is as follows:

- Find the set of population solutions that are worse than at least one offspring solution and denote the set by *S'*.
- If |*S'*| ≥ |S_0|, where S_0 is the set of offspring solutions, then apply SA to a proportion of solutions in S_0 and replace |S_0| worst population solutions with all members of S_0
- If |*S'*| < |S_0|, then apply SA to a proportion of |*S'*| best offspring solutions and replace |*S'*| worst population solutions with |*S'*| best offspring solutions.

4.3. Multi-objective Hybrid Clonal Selection Algorithm

The second algorithm mHCSA developed in this study is a kind of Clonal Selection Algorithm (CSA), a member of classes of metaheuristic algorithms which are inspired by the defense strategies used by the immune cells of the human body called Artificial Immune Systems. This class of algorithms has come into the focus of researchers during the recent decade. All the algorithms in this field are based on the clonal selection theory which is the foundational principle of immunology. The theory explains how the body's immune cells defeat the deceases and more importantly, how this complex system memorizes decease fighting experiences to use them in the future for the same or a range of similar deceases.

CSA was first proposed by De Castro and Von Zuben (2000; 2002), and now is one of the most popular immune based algorithms. The solution strategy adopted by the algorithm can be explained using technical terms of biological science. Note that the real immune systems' elements and their behavior are highly complicated and the concepts have been simplified to a great degree to be embedded in the algorithm. The algorithm maintains a set of solutions as the population of immune system's lymphocytes (B-cells). Each B-cell has a receptor called antibody that is used to defeat the foreign molecules which are named antigens. When the immune system is exposed to an antigen, a number of B-cells which their antibodies highly match the antigen (i.e., are most suitable to defeat it) become activated and are cloned to form a population of cells with high affinity antibodies. Affinity here means the matching degree of B-cell's antibody and the antigen. In the rest of the text, the term antibody will be used as the counterpart of B-cells'. After exposition to the antigen and defeating it, the cloned antibodies are subjected to affinity maturation or hypermutation. Hypermutation is a mutation process applied to the antibody, which is inversely dependent to the affinity level. The cloned and hypermutated antibodies are inserted in the population. Then, the receptor editing is performed by replacing a number of low affinity antibodies with entirely new antibodies. In receptor editing, a number of antibodies with low affinity are deleted and are replaced with entirely new antibodies. The repetition of the whole procedure of selection, cloning, hypermutation, and receptor editing by the immune system provides enhanced responsiveness of the body to the future infections.

In this paper, the CLONALG (CLONal selection ALGorithm) version of CSA is used. In the basic version of CLONALG, two population sets are maintained; one as the memory antibody repertoire, and the other as the remaining antibody repertoire. CSA CLONALG was initially developed to solve pattern recognition type problems. Antigens in these problems are known and are used as the input to the algorithm, but in combinatorial optimization problems, the antigen is not used directly as it is assumed as the unknown ideal solution which all antibodies are compared to it by their fitness values. Also, only one population is maintained for solving these problems, unlike the original version. Hypermutation is any mutation method that perturbs the solution. In the algorithm developed here, a short version of SA substitutes the hypermutation, so that the result is a hybrid algorithm of CLONALG and SA. A hybrid version of CLONALG and SA was previously proposed by Zhong *et al*. (2005; 2006) to solve pattern recognition problems. In their algorithm, SA is used as an improvement search process, which is applied on the hypermutated solutions to further improve them. In the algorithm developed here, SA is used as a direct substitute of hypermutation. As will be explained later, SA is customized to be dependent to the affinity level of a solution. In the following, the general steps of the algorithm are presented.

Phase 1: Generate an initial population of *NP* solutions:

- Repeat the following until *NP* solutions are generated:
    - Generate a solution using the initial solution generation procedure.
    - Check the road network connectivity, and the feasibility of the round trip time of each bus line.
    - If the solution is feasible, add it to the population; otherwise discard it.
- Compute the *S* values for population solutions.
- Build the initial Pareto-optimal set from the initial population.
- Compute fitness values of population and Pareto-optimal solutions.

Phase 2: Repeat the following procedure for *T* iterations:

- Select *NC* solutions with highest affinity levels (lowest fitness values).
- Clone each selected solution by replicating it *Beta*×*NP* times.
- For each solution in the produced clone, repeat:
    - Hypermutate the solution by applying SA.
    - Compute its fitness value.
- Compose the new population using the best solutions among the clone and the current population solutions.
- Update the Pareto-optimal set using all solutions generated by SA.
- Recalculate *S* values and fitness values of population and Pareto-optimal set.
- Select *RE* population solutions with highest fitness values, and replace them with new randomly generated solutions (receptor editing).
- Update the Pareto-optimal set again using the newly inserted solutions to the population.
- Recalculate *S* values and fitness values of population and Pareto-optimal set.

Hypermutation Process

As noted earlier, SA is used in the hypermutation step. The extent and scale of hypermutation is inversely proportional to the affinity level of the solution In other words, the extent and scale of the changes on the selected solution using SA must be higher for lower affinity degree solutions, and vice versa. Thus, not all the solutions in the clone must be undergo the same hypermutation process in the SA algorithm. Here, different combinations of move generation methods are used to reflect different degrees of changes caused by SA.

In general, three types of move generation methods are used; (a) a sub-routine to change the bus network topology, (b) a sub-routine to change the lane addition or link construction projects, and (c) a sub-routine to change the link lane allocations. The first and the last sub-routines are the same as is described for mHGA. The second sub-routine consists of swapping a present lane addition or link construction project with a new project, subject to the conditions that the construction cost does not exceed the budget limit and the omission of the present project does not lead to disconnectivity of the road network or bus routes.

Each of the proposed sub-routines can produce different types and scales of change in the solution, and the scale of the change increases if they are used in combination. Definitely, the combination of (a) to (c) causes more changes than the combination of only two sub-routines, and this in turn causes more changes than single sub-routines. It is not easy to predict which of the three sub-routines produce greater changes, but it can be said that (a) is more likely to cause more changes in the solution than (b) and (c). Sub-routine (a) makes major changes by modifying bus routes, since besides the bus network it may transform some one-way links to two-way links in need of providing lanes for new bimodal arcs. Sub-routines (b) and (c) produce fewer changes compared to (a). Based on the above explanations, the combinations of sub-routines are divided into four categories: (1) all sub-routines, (2) sub-routines (a)-(b) or (a)-(c), (3) sub-routines (b)-(c), (4) sub-routine (a), (b) or (c).

Each solution is evaluated for its fitness value and one of the above combinations is decided to be used in the SA implemented. For this purpose, the interval between the normalized highest and lowest fitness values in the clone is divided into four equal length sub-ranges, each one corresponding to one of the above four categories. Then, the fitness value of each solution is checked to find the sub-range that it resides in, and the type of move to be used in SA is defined. If the minimum and maximum fitness values in the clone are denoted by $F_{\min}$ and $F_{\max}$, then the normalized interval can be obtained as $[F_{\min}/F_{\max}, 1]$ with its interval length being denoted by F'. The following set of rules is used to define the move type:

- For the values in $[F_{\min}/F_{\max}, F_{\min}/F_{\max}+0.25F')$, randomly select to apply one of sub-routines (a), (b) or (c).
- For the values in $[F_{\min}/F_{\max}+0.25F', F_{\min}/F_{\max}+0.5F')$, apply both sub-routines (b) and (c).
- For the values in $[F_{\min}/F_{\max}+0.5F', F_{\min}/F_{\max}+0.75F')$, randomly select to apply either the combination of sub-routines (a)-(b) or the combination of subroutines (a)-(c).

- For the values in [$F_{\min}/F_{\max}$+0.75F', 1], apply all subroutines.

The rest of SA attributes are similar to the one described in mHGA. Similarly, any infeasible solution (leading to a disconnected network or bus lines having infeasible round trip times) is discarded, and the procedure is repeated again.

5. Computational Results

5.1. Test instances and data

There is no benchmark instance to assess the performance of the algorithms, because of the novelty of the problem. Thus, a number of test examples from other related references in the field of NDPs and TNDPs were adapted and necessary attributes were included in them. These test examples are divided into three general types; three small networks, three medium networks, and one large network. The provided information in the given references were used and reasonable values were chosen for the additional required data such as lane addition and link construction projects, budget limit, the number of lanes in each street, bus lines and their frequencies, walking times for the bus mode, bus passenger capacities, etc. Construction costs are assumed to be linear functions of the lane numbers. The maximum allowable round trip times were set to be equal to or slightly greater than the round trip times of existing bus lines. The increased round trip times reflect the possible improvements on the size of the fleet.

The following values or ranges were used for the other parameters: $\pi = 2$; $\mu = 3$; $\psi = 0.2$ minute; $\theta = 0.45$ minute^{-1}; $\bar{t}_{\hat{i}}$ = (5, 10) minutes; $n_e = 40$; f_e = (3, 12) buses per hour; ρ_{ij} =0.4* t_{ij}^{0} monetary unit; $\hat{\rho}_{\tilde{i}\tilde{j}}$ =0.25 monetary unit; $VOT_t = 0.333$; $VOT_a = 0.333$; $VOT_w = 0.667$, all in monetary unit per minute; $\alpha = 0.15$; $\beta = 4$; $\alpha' = 0.1$ and $\beta' = 4$; $\hat{\alpha}$ = 0.1 and $\hat{\beta}$= 4. The monetary unit used in this paper is 1,000 Rials (Currency in Iran), which is about 0.096 USD. To limit computational complexity, all bus lines are considered as attractive lines and connectivity is only considered for OD pairs.

Table 2. Test examples and their attributes

Network size	**Network adopted**	**Notation**	**Nodes No.**	**Links No.**	**OD Pairs No.**	**Bus Lines No.**
Small	The Abdulaal and LeBlanc (1979) network	AL	9	12	(16)	(2)
	The Nguyen and Dupuis (1984) network used in Kov *et al.* (2010)	ND	13	19	4	2

	A reduced version of Sioux Falls network used in LeBlanc *et al.* (1975)	SF1	14	19	176	(4)
Medium	The Nagurney (1984) network	NA1	20	28	8	(5)
	The basic Sioux Falls network of LeBlanc *et al.* (1975) used in Abdulaal and LeBlanc (1979)	SF2	24	37	528	5
	The Nagurney (1984) network	NA2	25	37	6	(4)
Large	The Nagurney (1984) network	NA3	40	66	6	(5)

Table 2 depicts the attributes of test examples and their references. The data in the parentheses indicate that they are not given in the references and so they are added by the authors. The only needed data for bus lines are their terminal nodes. One of the bus lines in test instance ND was omitted as there were two bus lines, sharing the same terminal nodes.

5.2. Parameter setting

The parameters used in mHGA are the population size *NP*, the number of generations *G*, the number of offspring to be generated *NO*, the proportion of the offspring solutions that SA is applied on, and the SA parameters. The parameters of mHCSA include the population size *NP*, the number of iterations *T*, the number of high affinity solutions selected to be cloned *NC*, the cloning coefficient for each solution *Beta*, the proportion of population solutions selected for receptor editing *RE*, and the SA parameters. The parameter settings for mHGA were adopted from the previous works of the authors (Miandoabchi *et al*., 2010; 2011); except for the number of offspring generated. In the previous works, the special crossover operator generates *2*×*N* offspring solutions and thus is problem size dependent. However, in this paper, the number of offspring solutions generated is a proportion of *NP*. This value is set to 0.25×*NP*. The parameter values for mHCSA were set by using a series of experiments and by considering the number of generated solutions by mHGA. Table 3 depicts the parameter settings.

Table 3. Parameter settings for the algorithms

Algorithm	Main parameters	SA parameters
mHGA	***NP* (*Population size*)** = 60 ***G* (*Number of generations*)** = 200 ***NO* (*Number of offspring to generate*)** = 0.25×*NP* ***Proportion of solutions to apply SA*** = 1/4	***Start temperature*** T_{start}**:** 5 ***Reduction rate R:*** 0.85 ***Stop temperature*** T_{stop}**:** 1
mHCSA	***NP* (*Population size*)** = 50 ***T* (*Number of iterations*)** = 21	***Start temperature:*** 5 ***Reduction rate:*** 0.85

	***NC* (*Selected solutions number*)** = 0.1×*NP* ***Beta* (*Cloning coefficient*)** = 0.2 ***RE* (*Receptor editing number*)** = 5	***Stop temperature:*** 1

To have a fair comparison of the performance of the two algorithms, the total computational effort of the two algorithms is required to be as close as possible to each other. The maximum number of the solutions generated by each algorithm indicates the approximate total computational effort and hence can be used as a guideline for setting the parameters. Since the parameters of mHGA are set previously, the number of solutions generated by mHGA can be used as a guideline for parameter setting in mHCSA.

For mHGA, the number of solutions generated is equal to $G\times NO\times(1+1/4\times Itr)$, because for G generations, *NO* offspring are generated and evaluated once, and SA is applied on 1/4 of them for *Itr* iterations, where *Itr* is equal to $1+\left[\log\left(\frac{T_{stop}}{T_{start}}\right)\Big/\log\left(R\right)\right]$. The proportion 1/4 is chosen to deal with the trade-off between high computational effort and the solution quality. For mHCSA, the number is equal to $T\times\left((NC\times Beta\times NP)\times Itr+RE\right)$, because for *T* iterations, *NC* solutions are selected, and each is cloned to produce *Beta*×*NP* solutions, all clone solutions are subjected to hypermutation by SA with *Itr* iterations, and at the end *RE* new solutions are produced and are inserted in the population.

To set the parameter values for mHCSA, the combination of the parameter values must be chosen so that the above numbers are equal or very close to each other. Therefore, the parameter combinations used for experiments were chosen to comply with the above rule. The same SA parameter setting in mHGA was used for hypermutation.

5.3. Software and hardware

All algorithms were coded by Matlab R2007a and the tests were carried out on a laptop with a Core2Duo T7500 2.2GHz CPU, and a 2 GB RAM. Each algorithm was run 5 times for each test instance.

5.4. Performance Comparison

The multi-objective nature of the problem implies that the performance evaluation of the algorithms is not possible through the comparison of single objective function values. Since both algorithms produce the Pareto-optimal solution set, the dominance and non-dominance of the obtained solution set is used to investigate the quality of the sets. On the other hand, it is possible to check the best values of each single objective function to identify the extent of the ideal Pareto-optimal solution space that has been explored.

In this paper, two measures are used to evaluate the algorithms. The first measure M_1 give the size of the Pareto-optimal solution set, and the second one M_2 is a set coverage measure proposed by Zitzler *et al.* (2000). The second measure considers the pair-wise comparison of the algorithms in terms of the fraction of Pareto-optimal solutions obtained by one algorithm that dominates or is equal to at least one solution obtained from another algorithm. Mathematically, M_2 is defined as follows:

$$M_2(X_i) = \sum_{j\neq i} C(X_i, X_j) - \sum_{j\neq i} C(X_j, X_i) \tag{60}$$

where $C(X_i, X_j)$ is the measure proposed in Zitzler *et al.* (2000) and is calculated as follows:

$$C(X_i,X_j) = \frac{\left|\left\{a_j \in X_j;\, \exists a_i \succ= a_j\right\}\right|}{\left|X_j\right|} \tag{61}$$

where $a_i \succ= a_j$ means that solution a_i dominates or equals solution a_j, and $C(X_i, X_j)$ is not necessarily equal to 1- $C(X_j, X_i)$. The formula (56) calculates the fraction of solutions in set X_j that is dominated by or equal to at least one solution in X_i.

Table 4 depicts the summary of the average values of the two measures and, the average runtimes for each test example.

Table 4. Summary of computational results

Test example	**mHGA**			**mHCSA**		
	M_1	M_2	***Runtime****	M_1	M_2	***Runtime****
AL	133	0.95	1,960	123	0.03	2,078
ND	368	0.47	2,302	304	0.18	3,580
SF1	227	0.82	6,699	131	0.03	7,536
NA1	270	0.51	11,085	191	0.19	11,718
SF2	124	1	20,800	129	0.06	18,940
NA2	207	1	8,227	104	0.08	8,061
NA3	217	0.90	20,844	76	0.01	18,957

* Average runtime (in seconds)

It can be observed from the table that mHGA has the highest values for measure M_1 in all test instances or examples, except for test instance SF2. mHGA clearly outperforms mHCSA in all values of measure M_2. A comparison of runtimes reveals that mHGA obtains feasible solutions faster in small and medium test instances on average. Figure 6 illustrates the average runtimes of the algorithms. The values for each algorithm are relatively close to each other, since the total computational efforts have been set to be the same. The runtimes for SF2 are significantly higher than the runtimes for NA1 and NA2, which has similar number of nodes and links. This can be explained by the higher number of OD pairs which requires a larger number of shortest path examinations, higher computational efforts for the equilibrium flow calculation and the most important, a larger number of arcs in the modified bus networks.

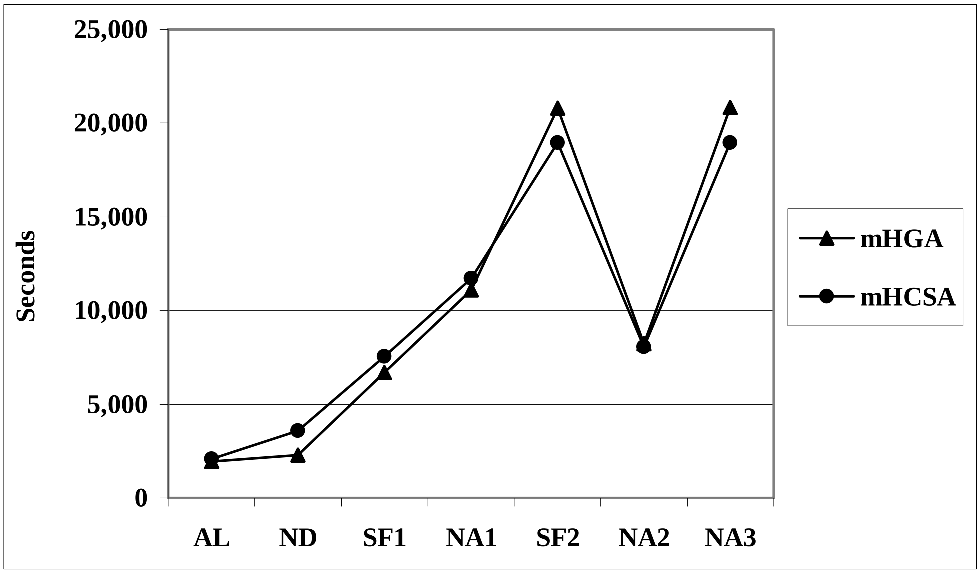

Figure 6. Average runtimes

Another performance comparison can be made by evaluating the best values for each single objective function in the Pareto-optimal sets obtained by each algorithm. The best values for each single objective function among all 5 runs in the Pareto-optimal sets of each algorithm are given in Table 5.

Table 5. The best objective function values

Test	mHGA				mHCSA			
example	Z_1	Z_2	Z_3	Z_4	Z_1	Z_2	Z_3	Z_4
AL	171,580	0.167	1	8.7	141,200	0.052	1	8.8
ND	5,477,700	0.599	1	13.7	2,802,700	0.607	1	13.4
SF1	843,810	0.543	1	4.1	725,720	0.542	1	4.0
NA1	722,980	0.492	1	34.9	801,580	0.492	1	36.7
SF2	4,956,200	0.529	1	5.5	4,076,000	0.521	1	4.9
NA2	519,070	0.492	1	35.9	463,560	0.492	1	33.5
NA3	237,840	0.495	1	47.1	199,250	0.494	1	49.3

An investigation of the values in table 5 reveals that mHGA has reached 6 best and 1 second best values for Z_1, 4 best and 1 second best values for Z_2, and 3 best and 4 second best values for Z_4. mHCSA has reached 1 best and 6 second best values for Z_1, 1 best and 4 second best values for Z_2, and 4 best and 3 second best values for Z_4. Both algorithms have obtained similar values for Z_2 for test examples NA1 and NA2. In conclusion, the results in both tables suggest that mHGA outperforms mHCSA in terms of both measures M_1 and M_2, and can obtain the best values for each individual objective function.

Another comparison between mHGA and the classic GA without SA (mGA) was performed to illustrate the effect of hybridization on the performance of GA. For this purpose, SA in mHGA was replaced with a single perturbation as the mutation operator. The perturbation is similar to the one applied in one iteration of the embedded SA. The same parameter setting was used for mGA, except that *Itr* in $G \times NO \times (1+1/4 \times Itr)$ was set to be equal to 1 and *G* was set to be equal to 560 to ensure the computational effort for generating the new solutions to be the same as mHGA. mGA was solved for the small test examples. Table 6 depicts the comparison between the two GAs for the best individual objective function values and the performance measures M_1 and M_2.

Table 6. Summary of computational results for mHGA and mGA

Test example	M_1	M_2	Z_1	Z_2	Z_3	Z_4	***Runtime****
				mHGA			
AL	133	0.67	171,580	0.167	1	8.7	1,960
ND	368	0.40	5,477,700	0.599	1	13.7	2,302
SF1	227	0.41	843,600	0.541	1	4.1	6,699
				mGA			
AL	242	0.29	167,050	0.075	1	8.8	3,016
ND	521	0.23	5,244,500	0.538	1	11.4	2,142
SF1	224	0.37	870,260	0.541	1	4.0	10,889

* Average runtime (in seconds)

The results in table 6 indicate that hybridizing GA with SA significantly improves the performance of GA, both in terms of measures M_1 and M_2, and the best values of individual objective functions. It can be observed that mHGA outperforms mGA in nearly all three test examples, except for Z_1 and Z_4 in test example SF1, and Z_4 in test example ND.

6. Conclusions and future research

In this paper, a multi-objective bimodal urban transportation network design problem is investigated, which includes automobile and bus modes. The problem consists of simultaneously designing road and bus networks, and considers the influences of flows of both modes on each other. The decisions considered for the road network are: constructing new streets, adding lanes to the existing streets wherever it is possible and benefical, determining the directions and locations of one-way streets, and determining lane allocations in two-way streets. The design of the bus network is performed by keeping the terminal stations of the existing bus lines unchanged and redesigning the forth and back routes of each existing bus line. The main contribution of the paper is proposing a new road network design problem with the inclusion of bus network decisions. This is based on the fact that the changes in road network design and bus network topology cannot be made without taking into account the effects of auto and bus flows on each other, and that street lane allocations and street directions cause restrictions on the bus network and vice versa. The problem is modeled as a bi-level mathematical problem with equilibrium constraints in which the upper level problem corresponds to the network authority's problem, and the lower level problem corresponds to the modal-split/assignment problem.

Because of the intrinsic complexity and non-convexity of the model, two hybrid multi-objective metaheuristic algorithms are developed to solve the problem. The proposed algorithms are the hybrid genetic algorithm, and the hybrid clonal selection algorithm. The algorithms generate a set of Pareto-optimal solutions using a fitness value calculation method previously proposed for a famous multi-objective genetic algorithm. The algorithms are evaluated by solving test instances taken from the literature and customizing them for the problem on hand, and their performances are compared to each other using two types of measures, and the best values obtained for each individual objective function. The computational tests indicate that the proposed hybrid genetic algorithm outperforms the hybrid clonal selection algorithm.

This paper proposes a new research field for urban transportation network design, which is mostly suitable for urban areas in which government plays the major role in designing the bus network. For instance, another problem can be considered by allowing the entire topology of the bus network including the terminal node pairs and forth and back routes to be determined endogenously or by considering the allocation of exclusive bus lanes throughout the entire network.

Acknowledgements

The authors are grateful for the two anonymous referees and Prof. S.V. Ukkusuri for their constructive comments. The research was jointly supported by a grant (200902172003) from the Hui Oi Chow Trust Fund and two grants (201001159008 and 201011159026) from the University Research Committee of the University of Hong Kong.

References

Abdulaal M., LeBlanc L.J. (1979) "Methods for Combining Modal Split and Equilibrium Assignment Models", *Transportation Science* 13(4):292-314.

Bellei G., Gentile G., Papola N. (2002) "Network Pricing Optimization in Multi-user and Multimodal Context with Elastic Demand", *Transportation Research part B* 36(9):779-798.

Ben-Ayed O., Boyce D.E., Blair C.E. (1988) "A General Bilevel Linear Programming Formulation of the Network Design Problem", *Transportation Research Part B* 22(4):311–318.

Cantarella G.E., Pavone G., Vitetta A. (2006) "Heuristics for Urban Road Network Design: Lane Layout and Signal Settings", *European Journal of Operational Research* 175(3):1682–1695.

Cantarella G.E., Vitetta A. (2006) "The Multi-criteria Road Network Design Problem in an Urban Area", *Transportation* 33(6):567-588.

Ceder A., Wilson N.H.M. (1986) "Bus Network Design", *Transportation Research Part B* 20(4):331-344.

Chiou S. (2008) "A hybrid Approach for Optimal Design of Signalized Road Network", *Applied Mathematical Modelling* 32(2):195-207.

Clegg J., Smith M., Xiang Y., Yarrow R. (2001) "Bilevel Programming Applied to Optimising Urban Transportation", *Transportation Research Part B* 35(1):41-70.

D'Acierno L., Gallo M., Montella B. (2005) "A Multimodal Approach for Managing Transportation Design Problems of Real Size Networks", *Advanced OR and AI Methods in Transportation,* Eds: Jaszkiewicz A., Kaczmarek M., Zak J., Kubiak M., Publishing House of Poznan University of Technology, 200-205.

D'Acierno L., Gallo M., Montella B. (2006) "Optimisation Models for the Urban Parking Pricing Problem", *Transport Policy* 13(1):34-48.

Dafermos S. (1972) "The Traffic Assignment Problem for Multiclass-User Transportation Networks", *Transportation Science* 6(1):73-87.

Dafermos S. (1982) "Relaxation Algorithms for the General Asymmetric Traffic Equilibrium Problem", *Transportation Science* 16(2):231-240.

Dantzig G.B., Harvey R.P., Lansdowne Z.F., Robinson D.W., Maier S.F. (1979). "Formulating and Solving the Network Design Problem by Decomposition", *Transportation Research Part B* 13(1):5-17.

De Castro L.N., Von Zuben F.J. (2000) "Clonal Selection Algorithm with Engineering Applications", *Proceedings of Genetic and Evolutionary Computation Conference 00*, 36-37.

De Castro L.N., Von Zuben F.J. (2002) "Learning and Optimization Using the Clonal Selection Principle", *IEEE Transactions on Evolutionary Computation* 6(3):239-250.

De Cea J., Fernández J.E. (1993) "Transit Assignment for Congested Public Transport Systems: An Equilibrium Model", *Transportation Science* 27(2):133-147.

Desaulniers G., Hickman M.D. (2007) "Public Transit", In *Handbook in Operations Research and Management Science*, *Transportation*, Volume 14, Eds. Barnhart C., Laporte G., Elsevier. Amsterdam, 69-127.

Drezner Z., Salhi S. (2000) "Selecting a Good Configuration of One-Way and Two-Way Routes Using Tabu Search", *Control and Cybernetics* 29(3):725–740.

Drezner Z., Salhi S. (2002) "Using Hybrid Metaheuristics for the One-Way and Two-Way Network Design Problem", *Naval Research Logistics* 49(5):449-463.

Drezner Z., Wesolowsky G.O. (1997) "Selecting an Optimum Configuration of One-Way and Two-Way Routes", *Transportation Science* 31(4):386–394.

Drezner Z., Wesolowsky G.O. (2003) "Network Design: Selection and Design of Links and Facility Location", *Transportation Research Part A* 37(3):241-256.

Fernández J.E., De Cea J., Malbran R.H. (2008) "Demand Responsive Urban Public Transport System Design: Methodology and Application", *Transportation Research Part A* 42(7):951-972.

Ferrari P. (1999) "A Model of Urban Transport Management", *Transportation Research Part B* 33(1):43–61.

Florian M., Spiess H. (1982) "The Convergence of Diagonalization Algorithms for Asymmetric Network Equilibrium Problems", *Transportation Research Part B* 16(6):477-483.

Friesz T.L. (1985) "Transportation Network Equilibrium, Design and Aggregation: Key Developments and Research Opportunities", *Transportation Research Part A* 19(5-6):413-427.

Friesz T.L., Anandalingam G., Mehta N.J., Nam K., Shah S.J., Tobin R.L. (1993) "The Multiobjective Equilibrium Network Design Problem Revisited: A Simulated Annealing Approach", *European Journal of Operational Research* 65(1):44-57.

Gallo M., D'Acierno L., Montella B. (2010) "A Meta-Heuristic Approach for Solving the Urban Network Design Problem", *European Journal of Operational Research* 201(1):144-157.

Gao Z., Sun H., Shan L.L. (2004) "A Continuous Equilibrium Network Design Model and Algorithm for Transit Systems", *Transportation Research Part B* 38(3):235-250.

Guihaire V., Hao J.K. (2008) "Transit Network Design and Scheduling: A Global Review", *Transportation Research Part A* 42(10):1251-1273.

Hamdouch Y., Florian M., Hearn D.W., Lawphongpanich S. (2007) "Congestion Pricing for Multimodal Transportation Systems", *Transportation Research Part B* 41(3):275-291.

Holland J.H. (1975) *Adaptation in Natural and Artificial Systems,* The University of Michigan Press, Michigan, USA.

Jeon K., Lee J.S., Ukkusuri S.V., Waller S.T. (2006) "Selectorecombinative Genetic Algorithm to Relax Computational Complexity of Discrete Network Design Problem", *Transportation Research Record: Journal of the Transportation Research Board* 1964:91-103.

Kov M., Fukuda D., Yai T. (2010) "Frequency Design of Bus Transit Mixed Urban Transport Network", *Asian Transport Studies* 1(4):352-367.

LeBlanc L.J. (1975) "An Algorithm for the Discrete Network Design Problem", *Transportation Science* 9(3):183-199.

LeBlanc L.J., Morlok E.K., Pierskalla W.P. (1975) "An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem", *Transportation Research* 9:309-318.

Lee C.K., Yang K.I. (1994) "Network Design of One-Way Streets with Simulated Annealing", *Papers in Regional Science* 73(2):119-134.

Lo H.K., Yip C.W., Wan K.H. (2003) "Modelling Transfer and Non-linear Fare Structure in Multi-modal Network", *Transportation Research Part B* 37(2):149–170.

Lo H.K., Szeto W.Y. (2009) "Time-dependent Transport Network Design under Cost-Recovery", *Transportation Research Part B* 43(1), 142-158.

Magnanti T.L., Wong R.T. (1984) "Network Design and Transportation Planning: Models and Algorithms", *Transportation Science* 18(1):1–55.

Miandoabchi E., Zanjirani Farahani R., Szeto W.Y. (2010) "Multi-objective Discrete Urban Road Network Design", submitted to *Journal of Operational Research Society*.

Miandoabchi E., Zanjirani Farahani R., Szeto W.Y. (2011) "Bi-objective Bimodal Urban Road Network Design Using Hybrid Metaheuristics", *Central European Journal of Operations Research*, DOI 10.1007/s10100-011-0189-4.

Migdalas A. (1995) "Bilevel Programming in Traffic Planning: Models, Methods, and Challenge", *Journal of Global Optimization* 7(4):381-405.

Nagurney A. (1984) "Comparative Tests of Multimodal Traffic Equilibrium Methods", *Transportation Research Part B* 18(6):469-485.

Nguyen S., Dupuis C. (1984) "An Efficient Method for Computing Traffic Equilibria in Networks with Asymmetric Transportation Costs", *Transportation Science* 18(2):185–202.

Nguyen S., Pallottino S., Gendreau M. (1998) "Implicit Enumeration of Hyperpaths in Logit Models for Transit Networks", *Transportation Science* 32(1):54-64

Patil G.R., Ukkusuri S.V. (2007) "System-Optimal Stochastic Transportation Network Design", *Transportation Research Record: Journal of the Transportation Research Board* 2029:80-86.

Poorzahedy H., Abulghasemi F. (2005) "Application of Ant System to Network Design Problem", *Transportation* 32(3):251-273.

Poorzahedy H., Rouhani O.M. (2007) "Hybrid Meta-heuristic Algorithms for Solving Network Design Problem", *European Journal of Operational Research* 182(2):578-596.

Poorzahedy H., Turnquist M.A. (1982) "Approximate Algorithms for the Discrete Network Design Problem", *Transportation Research Part B* 16(1):45-55.

Roberts F.S., Xu Y. (1988) "On the Optimal Strongly Connected Orientations of City Street Graphs I: Large grids", *SIAM Journal of Discrete Mathematics* 1(2):199-222.

Roberts F.S., Xu Y. (1989) "On the Optimal Strongly Connected Orientations of City Street Graphs: II. Two East-West Avenues or North-South Streets", *Networks* 19(2):221-233.

Roberts F.S., Xu Y. (1992) “On the Optimal Strongly Connected Orientations of City Street Graphs: III. Three East-West Avenues or North-South Streets”, *Networks* 22(2):109-143.

Roberts F.S., Xu Y. (1994) “On the Optimal Strongly Connected Orientations of City Street Graphs IV: Four East-West Avenues or North-South Streets”, *Discrete Applied Mathematics* 49(1-3):331-356.

Sheffi Y. (1985) *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Spiess H., Florian M. (1989) “Optimal Strategies: A New Assignment Model for Transit Networks”, *Transportation Research Part B* 23(2):83-102.

Steenbrink P.A. (1974) “Transport Network Optimization in the Dutch Integral Transportation Study”, *Transportation Research* 8(1):11–27.

Szeto W.Y., Lo H.K. (2005) “Strategies for Road Network Design over Time: Robustness under Uncertainty”, *Transportmetrica* 1(1):47-63.

Szeto W.Y., Lo H.K. (2006) “Transportation Network Improvement and Tolling Strategies: The Issue of Intergeneration Equity”, *Transportation Research Part A* 40(3):227-243.

Szeto W.Y., Lo H.K. (2008) “Time-dependent Transport Network Improvement and Tolling Strategies”, *Transportation Research Part A* 42(2):376-391.

Szeto W.Y., Solayappan M., Jiang Y. (2011) “Reliability-based Transit Assignment for Congested Stochastic Transit Networks”, *Computer-Aided Civil and Infrastructure Engineering* 26(4):311-326.

Szeto W.Y., Jaber X.Q., O’Mahony M. (2010) “Time-dependent Discrete Network Design Frameworks Considering Land Use”, *Computer-Aided Civil and Infrastructure Engineering* 25(6) 411–426.

Szeto W.Y., Wu Y. (2011) “A Simultaneous Bus Route Design and Frequency Setting Problem for Tin Shui Wai, Hong Kong”, *European Journal of Operational Research* 209(2), 141-155.

Uchida K., Sumalee A., Watling D., Connors R. (2007) “A Study on Network Design Problems for Multi-modal Networks by Probit-based Stochastic User Equilibrium”, *Networks and Spatial Economics* 7(3):213–240.

U.S. Bureau of Public Roads (1964) *Traffic Assignment Manual*, U.S. Department of Commerce, Washington, D.C.

Ukkusuri S.V., Mathew T.V., Waller S.T. (2007) “Robust Transportation Network Design Under Demand Uncertainty”, *Computer-Aided Civil and Infrastructure Engineering* 22(1):6-18.

Van Nes R. (2002) *Design of Multimodal Transport Networks*, PhD Thesis, Delft University, Netherlands.

Yang H., Bell M.G.H. (1998) “Models and Algorithms for Road Network Design: A Review and

Some New Developments”, *Transport Reviews* 18(3):257–278.

Yang H., Wang J.Y.T. (2002) “Travel time Minimization Versus Reserve Capacity Maximization in the Network Design Problem”, *Transportation Research Record: Journal of the Transportation Research Board* 1783:17-26.

Ying J.Q., Yang H. (2005) “Sensitivity Analysis of Stochastic User Equilibrium Flows in a Bimodal Network with Application to Optimal Pricing”, *Transportation Research Part B* 39(9):769–795.

Zhang H., Gao Z. (2007) “Two-way Road Network Design Problem with Variable Lanes”, *Journal of Systems and Science and Systems Engineering* 16(1):50-61.

Zhong Y., Zhang L., Li P. (2005) “Multispectral Remote Sensing Image Classification Based on Simulated Annealing Clonal Selection Algorithm”, *Proceedings of IEEE International Geoscience and Remote Sensing Symposium* (*IGARSS '05*):3745-3748.

Zhong Y., Zhang L., Huang B., Li P. (2006) “An Unsupervised Artificial Immune Classifier for Multi/Hyperspectral Remote Sensing Imagery”, *IEEE Transactions on Geoscience and Remote Sensing* 44(2):420-431.

Zitzler E., Deb K., Thiele L. (2000) “Comparison of Multiobjective Evolutionary Algorithms: Empirical Results”, *Evolutionary Computation* 8(2):173-195.

Zitzler E., Laumanns M., Thiele L. (2001) “SPEA2: Improving the Strength Pareto Evolutionary algorithm”, Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.