CEJOR (2012) 20:583–621
DOI 10.1007/s10100-011-0189-4

ORIGINAL PAPER

Bi-objective bimodal urban road network design using hybrid metaheuristics

Elnaz Miandoabchi · Reza Zanjirani Farahani · W. Y. Szeto

Published online: 25 January 2011

Abstract In this paper a bimodal discrete urban road network design problem with bus and car modes is investigated. The problem consists of decision making for lane addition to the existing streets, new street constructions, converting some two-way streets to one-way streets, lane allocation for two-way streets, and the allocation of some street lanes for exclusive bus lanes. Two objectives are considered in the problem: maximization of consumer surplus, and maximization of the demand share of the bus mode. The interaction of automobile and bus flows are explicitly taken into account and a modal-split/assignment model is used to obtain the automobile and bus flows in the deterministic user equilibrium state. The main contribution of the paper lies in proposing a new network design problem that combines the road network design decisions with the decision making for bus networks. The problem is formulated as a mathematical program with equilibrium constraints. A hybrid of genetic algorithm and simulated annealing, a hybrid of particle swarm optimization and simulated annealing, and a hybrid of harmony search and simulated annealing are proposed to solve the problem. Computational results for a number of test networks are presented and investigated.

Keywords Bimodal network design · Multi-objective · Elastic demand · Hybrid metaheuristics · Exclusive bus lanes

E. Miandoabchi · R. Z. Farahani (✉)
Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran
e-mail: zanjiranireza@gmail.com

W. Y. Szeto
Department of Civil Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, People's Republic of China

1 Introduction

Planning to cope with the ongoing growth of travel demand and increasing volumes of traffic in urban transportation networks is one of the most challenging issues that government authorities often face with. Improvement plans for urban transportation networks usually encompass a wide range of schemes such as expansion and configuration of network of streets, adjustment of traffic control systems and road tolls, planning to expand public transit systems, and so on. The category of problems dealing with the design of road and public transit networks can generally be called the "Urban Transportation Network Design Problems" (UTNDPs). These problems can be divided into four distinct classes: the "Transit Network Design Problem" (TNDP), the "Road Network Design Problem" (RNDP), the "Bimodal Network Design Problem" (BMNDP), and the "Multi-Modal Network Design Problem" (MMNDP).

The TNDP deals with the design of public transportation networks and their attributes or parameters, and can even be broadened to include the operation of public transit modes. The public transportation network consists of a set of stations and links, which together form its skeleton. A set of routes are determined on the network and a group of public transportation vehicles operate on each route on a regular basis. In particular, the TNDP for the bus mode has been extensively studied by various scholars. A wide range of problem assumptions and modeling approaches exist in the literature of the TNDP for the bus mode. Typical problems in this field are the design of network topology and operating lines, vehicle frequencies setting, transit network timetabling, bus scheduling, driver scheduling, and combinations of these problems. Two comprehensive reviews of these problems were given in Desaulniers and Hickman (2007) and Guihaire and Hao (2008).

The RNDP is normally referred to as the "Network Design Problem" (NDP) in the literature. RNDPs are always modeled as Stackelberg leader-follower games. Network (or the government) authorities act as the leaders and network users respond to the decisions made by the authorities. These problems can be expressed as bi-level programming formulations, where the upper level problem optimizes the network based on the existing constraints and the intended criteria set by the leader, and the lower level problem describes the reaction of the network users to the design scenarios determined in the upper level problem. RNDPs can be reformulated to single-level problems by expressing the lower level problems as equilibrium constraints.

RNDPs mainly deal with single transportation mode, in which all vehicles such as private automobiles, taxis, or even buses are considered as one mode. The problem was extensively reviewed by Magnanti and Wong (1984), Friesz (1985), Migdalas (1995), and Yang and Bell (1998). Traditionally, these problems are considered in two forms: the Discrete NDP (DNDP) and the Continuous NDP (CNDP). The DNDP, such as street construction and street orientation, is concerned with the discrete decision variables (e.g., Poorzahedy and Turnquist 1982; Wong and Yang 1997; Chen et al. 2002, 2007, 2010a; Chen and Yang 2004; Gao et al. 2005; Meng and Khoo 2008; Meng et al. 2008), and the CNDP, such as setting signal timing and toll, is concerned with continuous attributes of the network (e.g., Yin 2000; Meng et al. 2001; Meng and Yang 2002; Gao and Song 2002; Chiou 2008a,b, 2009a,b,c; Yin et al. 2009; Chen et al. 2010b). Another less investigated form is the Mixed Network

Design Problem (MNDP) which deals with both continuous and discrete decision variables.

One drawback of RNDPs is that they usually ignore the impact of the decisions on the performance of public transportation systems, especially public transit networks and their efficiencies. Some RNDPs even assume fixed travel demand, while actually travel demand is elastic and is dynamically divided between existing modes. In real world networks, the effect of the decisions for one transport-mode network (say the automobile network) on the others (say the bus network) cannot be easily ignored. One reason is that the alteration of two-way streets to one-way streets may result in a disconnected bus network. Another reason is that the improvement of the public transit services can make them more attractive to be used. This in turn can reduce the use of private vehicles and result in lower congestion in urban areas. The third reason is that the improvement of the automobile network can worsen the performance of the buses when they share the streets with automobiles, leading to reduction of travel demand for bus services. It is therefore important to consider more than one mode and elastic demand.

The BMNDP and MMNDP are concerned with the UTNDPs considering both transit and road networks simultaneously but the BMNDP just considers two modes and the MMNDP considers more than two modes. Demand for each mode is elastic in these two problems. These problems can take into account the effects of RNDP decisions on public transit systems and transit demands.

Past efforts focused on both general BMNDPs or MMNDPs (e.g., Clegg et al. 2001; Van Nes 2002) and specific problems such as pricing problems (e.g., Bellei et al. 2002; Huang 2002; Ying and Yang 2005; D'Acierno et al. 2006; Hamdouch et al. 2007), the link layout and signal setting problem (e.g., Cantarella and Vitetta 2006), 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). Two issues have not been received much attention in these design problems:

1. The allocation of exclusive lanes to specific modes. Indeed, the allocation of exclusive lanes to the modes such as buses, high occupancy vehicles (HOV's), carpools, and bicycles has been implemented in many real world urban road networks to improve the traffic conditions and to reduce the travel times for those modes. The allocation of exclusive bus lanes is a way to achieve more efficient bus services especially in congested urban road networks. By allowing buses traveling on exclusive lanes, they can move faster since they are not affected by the heavy auto flow in streets which cause long delays for buses. However, bus travel times for crossing intersections could be deeply affected by congestion on shared lanes. To date, some studies (e.g., Seo et al. 2005; Elshafei 2006; Li and Ju 2009) on this issue have been performed. These studies mainly focus on a selected corridor or at most evaluating a few scenarios of possible allocations. Only Mesbah et al. (2008) determine the optimal combination of exclusive bus lanes in a bimodal network using the bi-level programming approach.
2. Determining optimal street network configuration such as determining the best combination of one-way and two-way streets as well as the lane allocation in

two-way streets, while explicitly considering bus routes. Decisions for street network design and especially the street directions cannot be made properly without taking into account the existing bus routes. Problems may arise in situations that new one-way streets interfere with original bus routes, even if the new one-way streets do not intend to modify or redesign the bus routes. Moreover, exclusive bus lanes must be allocated carefully by considering the available street capacities to avoid congestion for auto flow. Furthermore, possible lane addition can be determined in a way that they contribute to the improvement of both public and private traffic flow conditions.

The concurrent allocation of exclusive bus lanes and determining optimal street network configuration is a practical issue that is faced by the urban road network planning authorities, and is often in the form of possible scenario evaluations. For instance, in Tehran, the capital of Iran, one of the main two-way corridors, Vali Asr Street, was transformed into a corridor for one-way normal traffic but was allocated two dedicated lanes for forth and back bus routes (Tehran Times 2009). In another case, it was decided to allocate exclusive bus lanes for some existing corridors and the streets network configuration were redesigned (Tehran Times 2010).

The problem can be approached by the optimization techniques for a complete network, rather than scenario evaluations for a restricted set of streets. When dealing with a large set of streets, other decisions such as lane allocations and street constructions can be included to further improve the network in seeking for synergic effects of combined decisions on the network performance.

Based on the above explanations, and motivated by the fact that the concurrent design of public and private transportation networks provides more robust and efficient networks, this paper addresses a BMNDP taking in account the allocation of exclusive bus lanes in combination with a set of RNDP decisions. Specifically, the problem considers the following decisions:

(i) Constructing new streets;
(ii) Adding lanes to the existing network streets;
(iii) Determining the directions of one-way streets;
(iv) Determining lane allocation in two-way streets, and
(v) Allocating exclusive street lanes for buses.

The first four decisions were considered in DNDPs previously by some authors in various forms. Constructing new streets as an individual decision and sometimes in combination with lane addition were investigated by some authors (e.g., Steenbrink 1974; LeBlanc 1975; Poorzahedy and Turnquist 1982; Poorzahedy and Abulghasemi 2005; Poorzahedy and Rouhani 2007). Lane addition was considered in the studies as a yes-no decision among other decisions such as road resurfacing, or improving the capacity of existing lanes (e.g., LeBlanc 1975). Street orientation and in some cases in combination with signal timings were under focus in a number of works (e.g., Roberts and Xu 1988, 1989, 1992, 1994; Lee and Yang 1994; Drezner and Wesolowsky 1997; Drezner and Salhi 2000, 2002; Cantarella et al. 2006; Gallo et al. 2010). Lane reallocation in streets in peak periods has been considered in Zhang and Gao (2007), and lane allocation and street orientation along with signal timing and parking lane allocation were investigated in Cantarella and Vitetta (2006). Finally, the combination of

street construction and mixed one-way and two-way street configuration was studied in Drezner and Wesolowsky (2003).

In a previous work of the authors (Miandoabchi et al. 2010), a multi-objective DNDP was proposed to consider the decisions (i)–(iv). In the present paper the previous work is extended to encompass decision (v). The inclusion of the latter transforms the problem into a network design problem that deals with the simultaneous design of public transit (bus) and private vehicle networks. The type of the problem studied here has not been considered in any research before. The most related research to the problem studied in this paper, is the work of Li and Ju (2009) where a dynamic multimodal traffic assignment model is used to evaluate the performance of the automobile and bus modes on a multimodal test network under two conditions: network with exclusive bus lanes on some bus routes, and network without any exclusive bus lanes.

All of the mentioned issues require a compromise between the performance measures of private and public transportation networks. Thus the proposed problem is formulated as a bi-objective nonlinear bi-level programming problem. The upper level problem is to maximize the consumer surplus (i.e., the total user benefit) and to maximize the demand share for bus services. The lower level problem is a combined modal-split/assignment problem to capture the elasticity of the travel demand of each mode. Because of the intrinsic complexity and non-convexity of the model, three multi-objective hybrid evolutionary algorithms are proposed as the solution methods to obtain a set of Pareto-optimal solutions to the problem. To summarize, this paper contributes the literature in two aspects:

1. proposing a new bi-objective bi-modal network design problem which includes the bus network design decisions in a DNDP, while explicitly considering the interactions between bus and car flows, and
2. developing three improved meta-heuristics for the problem.

The rest of the paper is organized as follows: In the next section the problem is defined. In Sect. 3, the notations and mathematical formulation of the problem are given. In Sect. 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 Sect. 6.

2 Problem definition

The problem under consideration is to design a bimodal urban road network under a bi-objective decision-making framework taking into account the effects of automobile flow on bus flow or vice versa. Before proceeding to the problem definition, a clear definition of the network elements is presented. Here, all types of streets are referred to as "links" (i.e., the counterpart of the term "edges" in graph studies). Each link consists of two arcs if it is two-way, and each link consists of only one arc if it is one-way. Each arc is characterized by a set of street lanes. It is obvious that if a movement is not allowed in one direction of a link, no arc will exist in that direction.

A number of "bus lines" operates on the road network. A bus line is identified by a sequence of road network arcs, in which a group of buses runs through them,

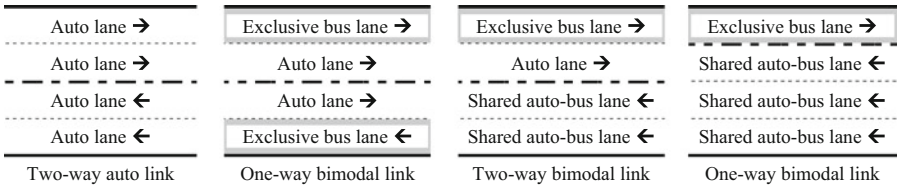

Fig. 1 Instances of lane allocations

and travels back and forth between two terminal stations. In this paper, bus lines are assumed to have forth and back routes on the same set of road network links. There are two reasons for this. First, this condition is imposed as there is an advantage that the passengers who get off at a bus stop in one direction can easily find another bus stop in the opposite direction for the return trip. Second, the TNDP literature (e.g., Abdulaal and LeBlanc 1979; Kov et al. 2010) usually does not distinguish between forth and back routes.

In this regard, the links of road networks can be distinguished as two types: (1) links that no bus lines pass through them are referred to as "auto links", and (2) links that bus lines pass through them are referred to as "bimodal links". Bimodal links may have lanes exclusively assigned to bus flow on one or both of their arcs. In the presence of exclusive bus lanes, buses are not allowed to travel on other lanes but travel on exclusive bus lanes, whereas in the absence of exclusive bus lanes, buses share the lanes with autos. Lanes are categorized in three types: (1) "auto lanes", i.e., lanes of arcs on auto links, or lanes of arcs on bimodal links in which exclusive bus lanes are allocated, (2) "exclusive bus lanes", and (3) "shared auto-bus lanes". Figure 1 depicts some instances of lane allocations.

2.1 Assumptions

The main assumptions for the studied problem are listed below:

- A basic network with all two-way links exists in advance where all bus routes share the street lanes with automobiles;
- Bimodal links are allowed to have different number of lanes in each direction, but two-way auto links must be symmetric in terms of lane allocation in each direction;
- Only one exclusive lane can be dedicated to buses on each arc of a link;
- Travel demand between each origin and destination (OD) pair is fixed and known, but the travel demand for each mode is elastic and can be determined by a logit modal split function;
- Travel times for crossing intersections are negligible;
- Auto and bus flows on shared lanes influence each other, and thus the travel time/cost function for each mode is asymmetric, and;
- 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., De Cea and Fernández 1993; Uchida et al. 2007).

Since we focus on network planning and not the actual operation, we need to propose a formulation that can handle large networks. The frequency-based approach is therefore more suitable than the schedule-based approach (e.g., Nuzzolo et al. 2001) in this case. Moreover, the frequency-based approach adopted by Uchida et al. (2007) can handle larger networks than that by Nguyen et al. (1998) as some of the line sections are combined in the modified network. Therefore, we adopt the modeling framework of Uchida et al. (2007) with some modifications to account for shared auto-bus lanes, exclusive bus lanes, and bimodal links.

2.2 Inputs

The following are the required inputs for the problem:

- The estimated total travel demand matrix;
- Existing bus lines and their frequencies;
- Attributes of network links such as capacities, investment costs, and travel time functions;
- Candidate new link construction projects defined by the network authority and their costs (indeed, a link construction project includes the construction of a new street connecting a pair of nodes of the network);
- Candidate lane addition projects and their costs (indeed, a lane addition project includes the construction of a pair/s of new lanes on each side of the existing street). The maximum possible number of lanes added for each project is determined by the network authority;
- Total 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 are outputs of the problem:

- The orientation of one-way links,
- The number of lanes allocated to each direction of links,
- New links constructed,
- New lanes added to the existing network links,
- Exclusive lanes allocated to the bus mode, and
- Auto and bus flows on each link.

Multi-modal network design problems are intrinsically more than one objective, since the design alternatives must be assessed based on various criteria of network users and the network authority. Here, two objective functions are considered for the problem. One is the maximization of consumer surplus and the other is the maximization of the passenger demand share of the bus mode. The first objective function is commonly used in UTNDPs and seeks to increase the benefit all users of the network on average regardless of the mode they use. On the other hand, the second objective function is incorporated in this paper to encourage more network travelers to use the

bus mode. Indeed, encouraging the use of public transport is a common objective of government authorities in many countries.

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 road network, where N is the set of intersection nodes, origin nodes, and destination nodes, and A is the set of existing network arcs. This network can be decomposed into an auto network and a basic bus network. The auto network is equivalent the existing road network whereas the basic bus network is defined by a set of binary variables stating whether bus lines pass through a road network link or not. In order to model the assignment of passengers in the bus network, in this study the concept of hyper-network modeling presented by Uchida et al. (2007) is adopted. Using this approach, the basic bus network is transformed into a modified bus network. The modified network $\hat{G}(\hat{N}, \hat{A})$ has a subset of road network nodes $\hat{N}$ to represent bus stations, and a set of arcs $\hat{A}$ to represent its route sections and walking arcs. A route section is defined by a portion of a route between two consecutive transfer nodes and it 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 (De Cea and Fernández 1993; Uchida et al. 2007). The walking arcs are introduced in the hyper-network to capture the walking time effect in the model and connect the bus stops in the modified bus network to origin or destination nodes in the expanded auto network where candidate new links are included. For ease of illustrating the concept, this paper assumes that the bus stops are near the heads or tails of arcs. This assumption is not far away from the reality as the bus stops in some cities like West Lafayette are located very near intersections. Nevertheless, we can add more dummy nodes to model the location of bus stops for other cases. Also, the walking distance related to using the auto mode is assumed to be small. We can add more nodes and arcs to capture walking time if it is not negligible. The following are the notations used in the model formulation.

3.1 Sets

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)
$\hat{A}$: set of modified bus network arcs $(\hat{i}, \hat{j})$
$\tilde{A}_{\hat{i}\hat{j}}$: set of road network arcs which affect the travel time on modified bus network arc $(\hat{i}, \hat{j})$
L: set of existing network links l
L': set of candidate new network links l'
S_l: set of arcs related to existing network link l

$S_{l'}$: set of arcs related to candidate new network link l'
W: set of all OD pairs (p, q)
E: set of bus lines e
$\hat{E}_{\hat{i}\hat{j}}$: set of bus lines passing through modified bus network arc $(\hat{i}, \hat{j})$
$\bar{E}_{\hat{i}\hat{j}}$: set of attractive bus lines passing through modified bus network arc $(\hat{i}, \hat{j})$
R_{pq}: set of auto routes r between OD pair (p, q)
$\hat{R}_{pq}$: set of bus routes $\hat{r}$ between OD pair (p, q)

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)
ω_{ij}: binary variable, which equals 1 if there is an exclusive bus lane on arc (i, j), and zero otherwise
X_r^{pq}: auto traffic flow on route $r \in R_{pq}$ (vehicles per hour)
$\hat{X}_{\hat{r}}^{pq}$: bus passenger flow on route $\hat{r} \in \hat{R}_{pq}$ (passengers per hour)

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
K_l: current number of lanes on existing link l
$K_{l'}$: number of lanes on candidate new link l'
Λ_{ij}: binary variable, which equals 1 if arc (i, j) belongs to a bimodal link, and zero otherwise
δ_{ijr}: binary variable, which equals 1 if arc (i, j) is on route r, and zero otherwise
$\hat{\delta}_{\hat{i}\hat{j}\hat{r}}$: binary variable, which equals 1 if arc $(\hat{i}, \hat{j})$ is on route $\hat{r}$, 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)
$\hat{\rho}_{\hat{i}\hat{j}}$: fixed bus fare for traveling between $\hat{i}$ and $\hat{j}$

3.4 Functions

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 bus passenger flow on arc $(\hat{i}, \hat{j})$ (passengers per hour)
$T_{p'q'}$: minimum travel time cost between node pair (p', q') for the auto mode
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)
$g_l(y_l)$: 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 flow
t^2_{ij}: travel cost function of the auto mode on arc (i, j) in the presence of bus flow
t^3_{ij}: travel cost function of the bus mode on arc (i, j) in the absence of auto flow
t^4_{ij}: travel cost function of the bus mode on arc (i, j) in the presence of auto flow
$\hat{t}_{\hat{i}\hat{j}}$: 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 $(\hat{i}, \hat{j})$
$\tilde{t}_{\hat{i}\hat{j}}$: access and waiting time cost function of bus passengers on arc $(\hat{i}, \hat{j})$
h_{pq}: minimum generalized travel cost of bus passengers between OD pair (p, q)
$\hat{h}_{pq}$: minimum generalized travel cost of auto between OD pair (p, q)
Z_1: first objective function—consumer surplus or total user benefit
Z_2: second objective function—the demand share of the bus mode

3.5 Mathematical model

The problem under study can be formulated as a Mathematical Programming model with Equilibrium Constraints (MPEC). Although the problem is represented as a single level model, by nature it is a bi-level model with the upper level problem depicting the decision making problem of the network authority, and the lower level problem depicting the network user behavior in choosing travel routes. The upper level problem is formulated as a bi-objective mixed integer mathematical model. The lower level problem includes equilibrium constraints expressed as Variational Inequality (VI) constraints. Mathematically, the problem can be formulated as follows:

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

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

subject to

$$\sum_{l\in L} g_l(y_l) + \sum_{l'\in L'} g'_{l'}(u_{l'}) \le B \tag{3}$$

$$0 \le y_l \le y_l^{\max} \quad \forall l \in L \tag{4}$$

$$\omega_{ij} + \omega_{ji} + k_{ij} + k_{ji} = K_l + 2y_l \quad \forall l \in L, (i,j) \in S_l \tag{5}$$

$$k_{ij} + \omega_{ji} \ge \Lambda_{ij} \quad \forall (i,j) \in A \cup A' \tag{6}$$

$$k_{ij} + \omega_{ji} \ge \Lambda_{ji} \quad \forall (j,i) \in A \cup A' \tag{7}$$

$$\omega_{ij} \le \Lambda_{ij} \quad \forall (i,j) \in A \cup A' \tag{8}$$

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

$$z_{ij} \le k_{ij} \quad \forall (i,j) \in A \cup A' \tag{10}$$

$$k_{ij} \le M z_{ij} \quad \forall (i,j) \in A \cup A' \tag{11}$$

$$k_{ij} \le k_{ji} + M(1 - z_{ji} + \Lambda_{ji}) \quad \forall (i,j) \in A \cup A' \tag{12}$$

$$k_{ji} \le k_{ij} + M(1 - z_{ij} + \Lambda_{ij}) \quad \forall (i,j) \in A \cup A' \tag{13}$$

$$z_{ij} + z_{ji} \ge 1 \quad \forall (i,j) \in A \tag{14}$$

$$z_{ij} + z_{ji} \ge u_{l'} \quad \forall l' \in L', (i,j), (j,i) \in S_{l'} \tag{15}$$

$$z_{ij} + z_{ji} \le M u_{l'} \quad \forall l' \in L', (i,j), (j,i) \in S_{l'} \tag{16}$$

$$T_{p'q'} \le M \quad \forall p', q' \in N \tag{17}$$

$$k_{ij}, y_l \ge 0, \text{ are integers} \quad \forall (i,j) \in A \cup A', l \in L \tag{18}$$

$$z_{ij}, \omega_{ij}, u_{l'} \in \{0,1\} \quad \forall (i,j) \in A \cup A', l' \in L' \tag{19}$$

$$\sum_{(i,j)\in A\cup A'} \Big[(1 - \Lambda_{ij} + \omega_{ij}) \left(t_{ij}^1 + \rho_{ij}\right) \left(x_{ij} - x_{ij}^*\right) + (\Lambda_{ij} - \omega_{ij}) \left(t_{ij}^2 + \rho_{ij}\right) (x_{ij} - x_{ij}^*) \Big] + \sum_{(\hat{i},\hat{j})\in \hat{A}} (\hat{t}_{\hat{i}\hat{j}} + \tilde{t}_{\hat{i}\hat{j}} + \hat{\rho}_{\hat{i}\hat{j}})(\hat{x}_{\hat{i}\hat{j}} - \hat{x}^*_{\hat{i}\hat{j}})$$

$$- \sum_{(p,q)\in \hat{W}} \left(\frac{1}{\theta} \ln \frac{\hat{d}^*_{pq}}{\bar{d}_{pq} - \hat{d}^*_{pq}} + \psi \right) \cdot (\hat{d}_{pq} - \hat{d}^*_{pq}) \ge 0 \quad \forall x_{ij}, \hat{x}_{\hat{i}\hat{j}}, \hat{d}_{pq} \in \Omega \tag{20}$$

$$\Omega = \left\{ \begin{array}{l} x_{ij}, \hat{x}_{\hat{i}\hat{j}}, \hat{d}_{pq} | \bar{d}_{pq} = d_{pq} + \hat{d}_{pq} \, \forall (p,q) \in W; \; d_{pq} = \pi \sum_{r\in R_{pq}} X_r^{pq} \, \forall (p,q) \in W; \\ x_{ij} = \sum_{(p,q)\in W} \sum_{r\in R_{pq}} X_r^{pq} \delta_{ijr} \forall (i,j) \in A \cup A'; \; \hat{d}_{pq} = \sum_{\hat{r}\in \hat{R}_{pq}} \hat{X}_{\hat{r}}^{pq} \forall (p,q) \in \hat{W}; \\ \hat{x}_{\hat{i}\hat{j}} = \sum_{(p,q)\in W} \sum_{\hat{r}\in \hat{R}_{pq}} \hat{X}_{\hat{r}}^{pq} \hat{\delta}_{\hat{i}\hat{j}\hat{r}} \, \forall (\hat{i},\hat{j}) \in \hat{A}; \; x_{ij} \le M z_{ij} \, \forall (i,j) \in A \cup A'; \\ \hat{d}_{pq} = 0 \, \forall (p,q) \in W \backslash \hat{W}; \; 0 \le \hat{d}_{pq} \le \bar{d}_{pq} \, \forall (p,q) \in \hat{W}; \; 0 \le d_{pq} \le \bar{d}_{pq} \forall (p,q) \in W; \\ X_r^{pq}, \hat{X}_{\hat{r}}^{pq} \ge 0 \forall r \in R_{pq}, \hat{r} \in \hat{R}_{pq}; \; \hat{W} = \left\{ (p,q) | \exists \hat{r} \in \hat{R}_{pq}, \; \hat{X}_{\hat{r}}^{pq} > 0 \right\} \end{array} \right\} \tag{21}$$

Expression (1) depicts the first objective of the problem, which is to maximize the users' benefit or the consumer surplus, assuming that the logit model is used for modal split. The parameters of the expression are explained later in Sect. 3.6. Expression (2)

depicts the second objective of the problem, which is to maximize the demand share of the bus mode in the network.

Constraint (3) imposes a budget limit to the total cost of lane addition and link construction. Constraint (4) limits the number of lanes added to existing network links. Constraint (5) ensures that the total number of lanes on all arcs of an existing link equals the number of lanes on that link. Constraints (6)–(7) assure the connectivity of bus routes through the arcs of bimodal links; that is, for each arc, there must be an exclusive bus lane or at least one shared auto-bus lane to guarantee that the bus routes do not become disconnected. Note that an additional constraint could be used to restrict some sequences of links (i.e., corridors) to have similar designs in terms of the existence or non-existence of exclusive bus lanes, for example; but here, it is preferred to consider the problem in the general form. Constraint (8) allows the allocation of exclusive bus lanes only when the bus mode is present. Constraint (9) is similar to constraint (5) but is for new links. Note that we assume that bus routes are fixed and buses do not use new links. Constraints (10)–(11) assure that no capacity is assigned to non-existing arcs. Constraints (12)–(13) imply that two-way auto links must be symmetric in terms of lane allocation for each arc of the link. In this paper, no restriction is imposed on the lane allocation of bimodal links, in order to provide more flexibility in designing links to achieve the two objectives. Constraint (14) assures that at least one arc is present for each existing link. Constraints (15)–(16) ensure that at least one arc is built for each new link. Constraint (17) 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., the constraint ensures that a route exists between every pair of nodes. Constraints (18)–(19) impose conditions to variables' domains.

Constraints (20)–(21) present the variational inequality formulations of the combined modal-split/assignment problem as the lower level problem, which determines the deterministic user equilibrium auto flows x^*_{ij}, bus passenger flows $\hat{x}^*_{\hat{i}\hat{j}}$, and travel demands d^*_{pq} and $\hat{d}^*_{pq}$ for the auto and bus modes respectively through determining path flows. Since the travel times of the auto and bus modes are interdependent and the effect of one mode on another is not symmetric, the problem cannot be formulated as an equivalent minimization program (e.g., Dafermos 1972; Abdulaal and LeBlanc 1979; Sheffi 1985). However, the problem can be stated as the finite-dimensional variational inequality (20) in terms of arc flows, subject to constraint set (21). Note that the bus passenger demand of an OD pair can only have a non-negative value when at least one bus route exists between that OD pair; i.e., only the travel demand between OD pairs with bus routes can use the bus mode, and for other OD pairs, no bus passenger demand exists. An additional OD pair set $\hat{W}$ is defined in (21) to consider the set of OD pairs with at least one bus route between them. OD pairs not belonging to this set are imposed to have zero bus demand values.

3.6 Modal split and travel cost functions

The travel demand for the bus and auto modes for each OD pair is defined by the logit formula, which is widely used as a mode split function. The demand for the bus

service is obtained by:

$$\hat{d}_{pq} = \overline{d}_{pq} \frac{1}{1 + e^{\theta(h_{pq} - \hat{h}_{pq} - \psi)}} \tag{22}$$

where h_{pq} and $\hat{h}_{pq}$ are the minimum generalized travel costs of bus passengers and autos for OD pair (p, q). ψ is the bus-specific constant, and θ is another parameter of the logit modal split function. h_{pq} and $\hat{h}_{pq}$ are obtained using the following relations:

$$h_{pq} = \sum_{(i,j) \in A \cup A'} \sigma_{pqij} \left[(1 - \Lambda_{ij} + \omega_{ij}) t^1_{ij} + (\Lambda_{ij} - \omega_{ij}) t^2_{ij} + \rho_{ij} \right] \tag{23}$$

$$\hat{h}_{pq} = \sum_{(\hat{i},\hat{j}) \in \hat{A}} \hat{\sigma}_{pq\hat{i}\hat{j}} (\hat{t}_{\hat{i}\hat{j}} + \tilde{t}_{\hat{i}\hat{j}} + \hat{\rho}_{\hat{i}\hat{j}}) \tag{24}$$

where σ_{pqij} and $\hat{\sigma}_{pq\hat{i}\hat{j}}$ are equal to 1 if arcs (i, j) and $(\hat{i}, \hat{j})$ are respectively on the shortest auto path and shortest hyperpath connecting OD pair (p, q) and are equal to zeros otherwise. Since bus and auto flows are assumed to be interdependent to each other, different travel cost functions for road network links under various cases are formulated for each mode. As stated in the assumptions, only the travel costs of passing the streets are modeled and the intersection crossing travel costs are ignored. Note that the travel costs for autos are defined on a per auto basis and the travel costs for buses are defined on a per passenger basis. The travel cost functions are described below:

- The travel cost function for autos on arcs of auto links or bimodal links with exclusive bus lanes is defined by the well-known BPR function (see Szeto and Lo 2005, 2006, 2008; Szeto et al. 2010) and the value of in-vehicle travel time VOT_t. The travel cost function for arc (i, j) is expressed as follows:

$$t^1_{ij} = t^0_{ij} \left(1 + \alpha \left(\frac{x_{ij}}{c_{ij}(k_{ij})} \right)^{\beta} \right) VOT_t \tag{25}$$

  where α and β are parameters. $c_{ij}(k_{ij})$ is the capacity of arc (i, j) which is assumed to be 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 links with shared auto-bus lanes is defined by Eq. (24) (see Uchida et al. 2007 for the corresponding travel time function):

$$t^2_{ij} = t^0_{ij} \left(1 + \alpha \left(\frac{\mu \overline{f}_{ij} + x_{ij}}{c_{ij}(k_{ij})} \right)^{\beta} \right) VOT_t \tag{26}$$

  where μ is a parameter to convert bus flow into equivalent auto flow in terms of their impact on road congestion. $\bar{f}_{ij}$ is the total frequency of buses on arc (i, j).

i.e., it is the number of buses passing through arc (i, j) per hour, which is computed by $\bar{f}_{ij} = \sum_{e \in E} \lambda_{ij}^{e} f_e$, where f_e is the frequency of bus line e; λ_{ij}^{e} is the binary variable, which equals 1 if bus line e passes through arc (i, j) and equals zero otherwise. Unlike (25), the flow used in (26) consists of the auto flow x_{ij} and a fixed bus flow $\mu \overline{f}_{ij}$. The bus flow is fixed as buses operate on fixed routes at fixed frequencies.

- The travel cost function of bus passengers where the buses travel on exclusive lanes is represented by Eq. (27):

$$t_{ij}^{3} = \hat{t}_{ij}^{0} \left(1 + \alpha \left(\frac{\mu \overline{f}_{ij}}{c_{ij}(1)} \right)^{\beta} \right) VOT_t \tag{27}$$

where $\hat{t}_{ij}^{0}$ is the free flow travel time for buses traveling on arc (i, j), and c_{ij} (1) is the capacity of an exclusive bus lane, which is equal to the capacity of a single lane on arc (i, j). Equation 27 is a special case of Eq. (26). Since the total bus frequency is fixed, the travel cost function has a constant value.

- The travel cost function of bus passengers on arcs of bimodal links without exclusive bus lanes is defined as follows:

$$t_{ij}^{4} = \hat{t}_{ij}^{0} \left(1 + \alpha \left(\frac{\mu \overline{f}_{ij} + x_{ij}}{c_{ij}(k_{ij})} \right)^{\beta} \right) VOT_t \tag{28}$$

The modified bus network requires specific cost functions for its arcs. Each arc is associated with an in-vehicle travel cost, an access and waiting time cost, and fare. The in-vehicle travel cost function determines the modified travel cost that captures the effect of in-vehicle congestion. This is done by incorporating the in-vehicle congestion effect into the in-vehicle travel cost on the modified bus network arcs. In this study, the function proposed by Uchida et al. (2007) for in-vehicle travel time is adopted. This function incorporates the values of in-vehicle travel time and waiting time to obtain the modified in-vehicle travel cost:

$$\tau_{\hat{i}\hat{j}}^{e} = \left[\sum_{(i,j) \in \tilde{A}_{\hat{i}\hat{j}}} \omega_{ij} t_{ij}^{3} + (1 - \omega_{ij}) t_{ij}^{4} \right] \left[1 + \hat{\alpha} \left(\frac{\chi_{\hat{i}\hat{j}}^{e} + \tilde{\chi}_{\hat{i}\hat{j}}^{e}}{f_e n_e} \right)^{\hat{\beta}} \right] \tag{29}$$

where $\hat{\alpha}$ and $\hat{\beta}$ are the parameters of the function, and n_e is the passenger capacity of buses of line e. The first squared bracket term is the in-vehicle travel cost on arc $(\hat{i}, \hat{j})$. The second squared bracket term is a BPR type function that considers the effect of in-vehicle congestion due to the total number of passengers who compete for the capacity of line e, $\tilde{\chi}_{\hat{i}\hat{j}}^{e}$ with the bus passenger flow of that line on arc $(\hat{i}, \hat{j})$, $\chi_{\hat{i}\hat{j}}^{e}$. The in-vehicle travel cost on arc $(\hat{i}, \hat{j})$ is obtained by finding the average value of the corresponding modified in-vehicle travel costs on bus lines using that arc as follows (Uchida et al. 2007):

$$\hat{t}_{\hat{i}\hat{j}} = \frac{\sum_{e \in \bar{E}_{\hat{i}\hat{j}}} \tau^{e}_{\hat{i}\hat{j}} f_e}{\sum_{e \in \bar{E}_{\hat{i}\hat{j}}} f_e} \tag{30}$$

The line passenger flow is obtained using the following relation (De Cea and Fernández 1993):

$$\chi^{e}_{\hat{i}\hat{j}} = \hat{x}_{\hat{i}\hat{j}} \frac{f_e}{\sum_{e \in \bar{E}_{\hat{i}\hat{j}}} f_e} \tag{31}$$

The access and waiting time costs associated with arc $(\hat{i}, \hat{j})$ is defined as:

$$\tilde{t}_{\hat{i}\hat{j}} = \bar{t}_{\hat{i}} \cdot VOT_a + \frac{60}{\sum_{e \in \bar{E}_{\hat{i}\hat{j}}} f_e} VOT_w + \alpha' \left(\frac{\hat{x}_{\hat{i}\hat{j}} + \tilde{x}_{\hat{i}\hat{j}}}{\sum_{e \in \bar{E}_{\hat{i}\hat{j}}} f_e n_e} \right)^{\beta'} VOT_w \tag{32}$$

where α' and β' are the parameters of the function. VOT_a is the value of walking time, and VOT_w is the value of (extra) waiting time. The first term is the average walking time cost to the bus station at node $\hat{i}$. The second 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). Like De Cea and Fernández (1993), Gao et al. (2004), Uchida et al. (2007), and Szeto et al. (2011), the last term denotes the extra waiting time cost due to the insufficient capacity of bus lines passing through arc $(\hat{i}, \hat{j})$ (or in-vehicle congestion). Because of the insufficient capacity, passengers may not be able to board the first arriving bus and hence their waiting time cost increases. This extra waiting time cost is modeled by a power function.

The set of attractive lines $\bar{E}_{\hat{i}\hat{j}}$ for each arc $(\hat{i}, \hat{j})$ is obtained by the following mathematical model (De Cea and Fernández 1993):

$$\min_{\{\upsilon_e^{\hat{i}\hat{j}}\}} \frac{1 + \sum_{e \in \hat{E}_{\hat{i}\hat{j}}} \tau^{e}_{\hat{i}\hat{j}} f_e \upsilon_e^{\hat{i}\hat{j}}}{\sum_{e \in \hat{E}_{\hat{i}\hat{j}}} f_e \upsilon_e^{\hat{i}\hat{j}}} \tag{33}$$

subject to

$$\upsilon_e^{\hat{i}\hat{j}} \in \{1, 0\}, \quad \sum_{e \in \hat{E}_{\hat{i}\hat{j}}} \upsilon_e^{\hat{i}\hat{j}} \geq 1 \tag{34}$$

where $\upsilon_e^{\hat{i}\hat{j}}$ is the binary variable which defines the set of selected lines as attractive lines.

4 Solution procedures for the problem

Bi-level programming problems are indeed NP-hard problems. Ben-Ayed et al. (1988) showed that even a simple linear bi-level programming problem is still NP-hard. Many

bi-level programming problems are non-convex, and even when both upper and lower level problems are convex, there is no guarantee for the whole bi-level problems to be convex. Although Branch and Bound algorithms can be applied to solve for global optimal solutions to small-size bi-level problems, they are not able to solve real size problems. In recent years, there has been an increasing interest in solving UTNDPs by metaheuristic algorithms and especially their hybrid versions (e.g., Drezner and Salhi 2002; Poorzahedy and Rouhani 2007; Cantarella et al. 2006). The use of metaheuristic algorithms for solving other types of transportation related problems has also been among the subjects of interest (e.g., Reimann and Ulrich 2006; Sörensen 2006; Matis and Kohání 2010).

In our previous work (Miandoabchi et al. 2010), three hybrid multi-objective metaheuristics were proposed: a hybrid of genetic algorithm and simulated annealing, an evolutionary simulated annealing, and a hybrid of artificial bee colony algorithm and simulated annealing. All these algorithms used the adaptive weight approach to calculate the fitness values to solve the multi-objective single modal design problem.

In this paper, three hybrid multi-objective population-based metaheuristic algorithms are proposed to solve the bimodal problem: A multi-objective Hybrid Genetic Algorithm (mHGA), a new multi-objective Hybrid Particle Swarm Optimization (mHPSO) algorithm, and a new multi-objective Hybrid Harmony Search (mHHS) algorithm. All of them are hybridized with Simulated Annealing (SA), and adopted a distance-based fitness value calculation method to deal with the multi-objectivity of the problem. Indeed, mHHS and mHPSO have never been applied to solve urban transportation network design problems before. In the following, we proceed with the common attributes and the general characteristics of the algorithms and then describe each algorithm in detail.

4.1 Similarities of the proposed algorithms

All proposed algorithms are similar in terms of solution encoding, Pareto-optimal set generation, calculation of fitness values, checking for strong connectivity, and initial population generation.

4.1.1 Pareto-optimal set generation

Like many multi-objective metaheuristics, a set of Pareto-optimal solutions is defined to maintain the list of non-dominated solutions in the algorithms. A solution is added to the set after being checked for non-dominance by other solutions in the set and the dominated solutions are omitted from the set if necessary. The non-dominance of the new solution O' is verified by checking whether no solution in the set dominates O', where the solution O_s in the set is said to dominate O' if $Z_i(O_s) \geq Z_i(O'), i = 1, 2$ and $Z_i(O_s) > Z_i(O')$ for at least one objective. If no solution in the set is found to dominate O', then O' is considered as a Pareto-optimal solution. As will be described later, in the developed algorithms every newly generated solution is checked for Pareto-optimality.

4.1.2 Calculation of fitness values

As pointed out earlier, the fitness values are defined using the rankings obtained by the distance-based fitness calculation method. This method was proposed by Osyczka and Kundu (1995), and used by Chen et al. (2006) for a multi-objective CNDP. The method works with the relative distance between a generated solution and the members of the Pareto-optimal set. According to Chen et al. (2006), the basic idea is to assign a fitness value to each solution according to the distance measure with reference to the existing non-dominated solutions obtained in the previous generation, and a higher fitness value is assigned to a solution if it is farther away from the existing non-dominated solution set.

Each solution is given a fitness value and an additional value called the potential value. The latter is used for calculating fitness values of the new solutions, where the fitness values are determined based on calculating the relative distances of the new solutions to the nearest Pareto-optimal solutions obtained in the last generation, the potential values of these Pareto-optimal solutions, and the maximum potential values $P_{\max}$ as shown later.

The first generated solution of the algorithm is taken as a solution of the Pareto-optimal set with the potential value equal to the parameter d_1 (where its proper value is determined by tuning) and the fitness value of this solution equal to its potential value.

For each new solution O', the following procedure is repeated to compute its fitness value F and potential value:

- Calculate the relative distance to each existing member O_s of the Pareto-optimal set:

$$d_s(O') = \sqrt{\left(\frac{Z_1(O') - Z_1(O_s)}{Z_1(O')}\right)^2 + \left(\frac{Z_2(O') - Z_2(O_s)}{Z_2(O')}\right)^2} \tag{35}$$

- Find the minimum value of all relative distances:

$$d_{s^*}(O') = \min\left\{d_s(O')\right\} \tag{36}$$

  where s^* is the nearest Pareto-optimal set member to the solution O'.
- If O' is a new non-dominated solution which dominates at least one of the existing Pareto-optimal set members, calculate the fitness value of O':

$$F = P_{\max} + d_{s^*}(O') \tag{37}$$

  Update the Pareto-optimal set, set $P_{\max} = F$, and set the potential value of O' to be F.
- If O' is a new non-dominated solution but does not dominate any existing member in the Pareto-optimal set, calculate its fitness value of O':

$$F = P_{s^*} + d_{s^*}(O') \tag{38}$$

Add the new solution to the Pareto solution set with the potential value of O' equal to F. If

$$F > P_{\max} \text{ then set } P_{\max} = F.$$

- If O' is not a new Pareto-optimal solution, calculate its fitness value as follows:

$$F = P_{s*} - d_{s*}(O') \tag{39}$$

If $F < 0$, then set $F = 0$ (The fitness value must be non-negative).

The values for the objective functions Z_1 and Z_2 are respectively calculated from the minimum generalized travel costs and the demand share to the bus mode for each OD pair that are obtained by solving the lower level combined modal-split/assignment problem (20)–(21) 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 (20)–(21) 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 (33)–(34) 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" which is widely used as the solution technique of traffic assignment problems (e.g., Sheffi 1985, p. 223). The method consists of iteratively solving a series of equilibrium sub-problems. For each iteration of the diagonalization algorithm, the sub-problem with the following objective function to be solved subject to constraint set (21) (see Sheffi 1985, p. 239):

$$\begin{aligned}\text{Min } z = & \sum_{(i,j)\in A\cup A'} \int_0^{x_{ij}} \left[(1-\Lambda_{ij}+\omega_{ij})t_{ij}^1(w) + (\Lambda_{ij}-\omega_{ij})t_{ij}^2(w) + \rho_{ij}\right] dw \\ & + \sum_{(\hat{i},\hat{j})\in \hat{A}} \int_0^{\hat{x}_{\hat{i}\hat{j}}} \left[\hat{t}_{\hat{i}\hat{j}}(w) + \tilde{t}_{\hat{i}\hat{j}}(w) + \hat{\rho}_{\hat{i}\hat{j}}\right] dw + \sum_{(p,q)\in W} \int_0^{\hat{d}_{pq}} \left(\frac{1}{\theta} \ln \frac{w}{\bar{d}_{pq}-w} + \psi\right) dw\end{aligned} \tag{40}$$

4.1.3 Checking for strong connectivity

The strong connectivity of a road network is checked in two stages. At the first stage, all nodes are checked to have at least one outgoing and one incoming lane which acts as the necessary condition for the network strong connectivity. If the necessary condition holds, it is attempted to find the shortest paths between all OD pairs. If at least one OD pair is disconnected, i.e., no shortest path can be found between them, the created network is rejected. If the necessary condition does not hold, then the network

is rejected without the need of using any shortest path algorithm for checking. This procedure helps to avoid the unnecessary use of time-consuming shortest path algorithm especially for medium and large sized networks. Here, Dijkstra's shortest path algorithm is applied to find the shortest path between each node for the road network. Another type of strong connectivity needs to be considered, which is the connectivity of bus routes as stated earlier, to guarantee the back and forth movement of buses on the same routes.

4.1.4 Initial population generation

Initial population members are constructed randomly based on a heuristic procedure. First, the possible lane addition or link construction projects are randomly selected, until the construction cost reaches the budget limit. Then, the added lanes and existing lanes are randomly allocated to each arc taking into account two issues: (1) to generate as fewer disconnected networks as possible, and (2) to maintain the connectivity of bus routes. In order to reduce the possibility of generating disconnected networks, at the beginning, lanes are allocated in such a way that each node has at least one outgoing lane and one incoming lane. This is straightforward for auto links, but in the case of bimodal links, we must decide whether to allocate exclusive bus lanes or not, taking into account the connectivity of bus routes. For this purpose, exclusive bus lanes or shared auto-bus lanes are checked and added to each arc of those links if necessary. Depending on the number of remaining lanes to be allocated to a bimodal link and the current lane configuration of its arcs, several alternatives may exist for this link (e.g., 1 exclusive bus lane on each side, or 1 exclusive bus lane on one arc and 1 shared auto-bus lane on the other arc, or 1 shared auto-bus lane on each side). Then, one alternative is randomly chosen among all possible alternatives. Next, the remaining lanes are randomly allocated to the network. Any disconnected network generated is discarded and the procedure is repeated.

4.2 Comparison of the proposed algorithms

To provide an overall view to the developed algorithms, and to show their similarities and differences between these algorithms clearly, their general structural characteristics are compared in Table 1. The details will be described later.

4.3 Multi-objective hybrid genetic algorithm

Genetic algorithm is a population-based metaheuristic inspired by the genetic evolution process of species in the nature, which was first introduced by Holland (1975). GA has been successfully applied in the network design problems (e.g., Yin 2000, 2002; Drezner and Salhi 2002; Drezner and Wesolowsky 2003; Cantarella et al. 2006; Cantarella and Vitetta 2006; Chen et al. 2006, 2010a,b). The algorithm proposed here is a multi-objective hybrid version of GA and SA adopted from our previous work (Miandoabchi et al. 2010). This is adopted in this paper, since it outperformed the other two algorithms in our previous work. The algorithm used here differs from the

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
mHPSO	T iterations	Iteratively selecting each population solution, updating it using the PSO solution update mechanism, and applying SA	Replacing the selected solution with its updated and improved solution
mHHS	T iterations	Generating a new solution using either elements of population solutions or randomization, and applying SA	Replacing the worst population solution with the newly generated one

original version in some parts. For example, the proposed algorithm applies a different fitness value calculation method, and includes additional considerations related to the bus mode decisions. The whole procedure of the algorithm is presented below and the details are described later.

Phase 1: Generate a population of P solutions, compute their fitness values, and build the Pareto-optimal set from the initial population.

Phase 2: Repeat the following procedure for G generations:

- Select a pair of parent solutions:
  - Select the first parent from the population applying the roulette wheel rule.
  - Select the second parent randomly from the Pareto-optimal set.
- Merge the selected parents through a merging process to produce an offspring set.
- Check the connectivity of each offspring solution and discard the solution if the network is disconnected.
- Check the budget feasibility of each offspring solution and apply the budget reduction sub-routine if needed. If the infeasible solution cannot be repaired, then discard the offspring solution.
- Update the Pareto-optimal set and compute the fitness value for each 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 the k best offspring solutions selected, where k is a parameter, and update the population.

4.3.1 Solution encoding

The solutions are encoded as matrix chromosomes which represent the complete configuration of lane allocations in the network. Chromosomes are denoted as 4-rows matrices, in which columns correspond to network links and rows correspond to their

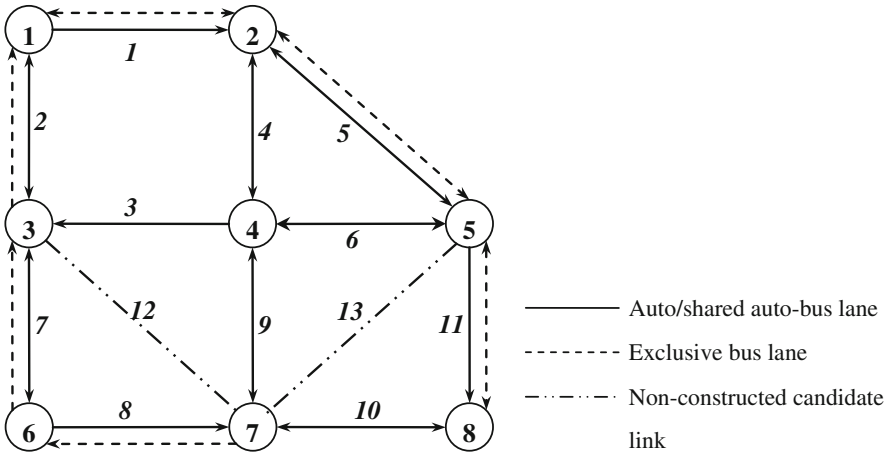

Fig. 2 Representation of a typical network

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

Fig. 3 Lane allocations matrix

lane allocations. The length of each chromosome is equal to $L + L'$, the total number of existing and candidate links. The first two rows of the chromosome denote the allocation of auto or shared auto-bus lanes to arcs (i, j) and (i, j) of link l or l', and the last two rows denote the allocation of exclusive bus lanes.

Figure 2 illustrates a typical 8-node and 11-link network, and Fig. 3 depicts its chromosome representation. The two dashed and dotted lines connecting nodes 3 and 7, and nodes 7 and 5 are the possible candidate links, which are not decided to be constructed in this network design scenario. Thus the chromosome representation will have 13 columns. All links consist of two or four lanes. Solid arrows represent the auto or shared auto-bus lanes, and dashed arrows represent exclusive bus lanes. One-head solid arrows denote one-way streets and two-head solid arrows denote two-way streets. One-head dashed arrows indicate the presence of exclusive bus lanes in the indicated direction, and two-head dashed arrows indicate the presence of two exclusive bus lanes, one for each direction. Two bus lines operate on the network: bus line 1 which operates on links 1–2, 2–5, and 5–8, and bus line 2 which operates on links 1–3, 3–6, 6–7, 7–4, and 4–5.

4.3.2 Crossover operator

The crossover operator used here is taken from Miandoabchi et al. (2010), which is similar to the version used in Drezner and Wesolowsky (1997, 2003) that can exploit

the structure of the problem. The crossover operator combines solutions in a way that the set of links taken from each network forms a connected set. Such a connected set is built using a "pivot" node. For a pivot node, a count is calculated for each link of the network. Count 1 is given to the links that are directly connected to the pivot node. Count 2 is given to the links which are connected to the links with count 1 and so on. Next, the median of all links is computed. Links with their counts below (above) the median are selected from the first (second) parent, and links with their counts equal to the median are taken randomly from one parent. Note that in the original version of the crossover operator in Drezner and Wesolowsky (1997, 2003), a link with its count equal to the median can be taken independently from either parents, but in this paper the whole set of such links are randomly taken from only one parent. The reason of using this new rule is to decrease the possibility of having disconnected solutions by limiting such links to be chosen from a single parent. After producing an offspring solution using the above method, the directions of one-way links with counts above the median are reversed to obtain a second offspring solution.

Each obtained offspring solution is examined for feasibility. If the solution is infeasible in terms of the connectivity of the network or the connectivity of the bus routes, the solution is discarded. Then the solution is checked for total construction cost and if the solution violates the budget limit, a budget reduction sub-routine is applied to repair the infeasible solution with main steps similar to the one used in Miandoabchi et al. (2010):

- Check every possible swap of lane addition or link construction projects; i.e., omission of an existing project and inclusion of another project.
- If there is at least one feasible swap after considering the budget limit and the network connectivity, select and apply a random swap.
- If there is no feasible swap, then check whether there are feasible omissions of the current projects in the solution.
- If there is at least one feasible omission in terms of network connectivity, then repeat the following step until the budget limit is not violated or the network becomes disconnected:
  - Select the maximum cost project and omit it.
  - If the omitted project belongs to an existing link, reallocate the lanes to maintain the previous arc directions, and in case of bimodal links, maintain the bus route connectivity too.
  - If the network becomes disconnected, then stop and report infeasibility.

4.3.3 Embedded SA algorithm

A short version of Simulated Annealing (SA) algorithm is applied on some of offspring solutions as is used in Miandoabchi et al. (2010) with some differences in defining random changes in lane allocation. This approach offers two advantages: First, it can substitute a mutation operator since it provides a series of random perturbations to the selected offspring solution. Second, it explores the neighborhood of the produced offspring solution in the hope of finding better solutions in the unexplored solution space. Because of the high computational requirements for large or even medium sized

problems, the SA algorithm is applied on a proportion of offspring solutions rather than all of them. The algorithm has the following characteristics:

- One iteration is performed per temperature.
- A random solution is generated by iteratively selecting each link and randomly deciding whether to apply a random change on its lane allocation or not.
  Changing the allocation of lanes may lead to a disconnected network in two ways: First, a link has zero outgoing lanes from or incoming lanes to the nodes. Second, no path connects between some pairs of nodes even when the first case does not occur. The second case is not easily predictable, but the first case can be avoided using special feasibility constraints. These constraints are used to find the best possible alternatives for lane allocation. Moreover, for bimodal links, the connectivity of bus routes must be taken into account.
- The feasibility constraints for an auto link is defined as follows:

$$k_{ij} + \sum_{p \in \Phi_i, p \neq j} k'_{ip} \geq 1 \tag{41}$$

$$k_{ji} + \sum_{p \in \Phi_i, p \neq j} k'_{pi} \geq 1 \tag{42}$$

$$k_{ij} + \sum_{p \in \Phi_j, p \neq i} k'_{pj} \geq 1 \tag{43}$$

$$k_{ji} + \sum_{p \in \Phi_j, p \neq i} k'_{jp} \geq 1 \tag{44}$$

$$k_{ij} + k_{ji} = k'_{ij} + k'_{ji} \tag{45}$$

where k'_{ij}, k'_{ji}, k'_{ip}, k'_{pi}, k'_{jp}, and k'_{pi} are the current values of the related lane allocation, and k_{ij} and k_{ji} are the lane allocation variables for link (i, j) (i.e., for arcs (i, j) and (j, i), respectively). Φ_i and Φ_j are correspondingly the sets of adjacent nodes to nodes i and j. The second terms in (40) and (43) calculate the total number of outgoing lanes to nodes i and j except the lanes of arcs (i, j) and (j, i), respectively. Similarly, the second terms in (41) and (42) calculate the total number of incoming lanes to nodes i and j except the lanes of arcs (i, j) or (j, i), respectively. The inequality pair (40)–(41) ensures that there are at least one outgoing lane from and one incoming lane to node i whereas the inequality pair (42)–(43) ensure that there are at least one outgoing lane from and one incoming lane to node j. Equation (44) is the lane allocation constraint, which ensures that the numbers of lanes before and after alteration are the same. Indeed, there are two alternatives for changing the lane allocation of each auto link. For a two-way auto link, the two alternatives are: (1) $k_{ij} = 0, k_{ji} = k'_{ij} + k'_{ji}$ and (2) $k_{ij} = k'_{ij} + k'_{ji}, k_{ji} = 0$. For a one-way auto link, one alternative is to change the direction of traffic flow completely (i.e., $k_{ij} = 0, k_{ji} = k'_{ij} + k'_{ji}$) and the second alternative is to change the one-way link to a two-way link (i.e., $k_{ij} = k_{ji} = (k'_{ij} + k'_{ji})/2$). For each alternative, the feasibility is checked via the constraint set (40)–(43) and any infeasible

alternative is omitted. Next, if more than one alternative remains, one alternative is chosen randomly; otherwise the remaining single alternative is chosen.

- In order to take into account the connectivity of bus routes on a bimodal link, the feasibility constraints (40)–(44) are modified through replacing k_{ij} by $k_{1ij} + k_{2ij}$, where the first term is the binary variable called "minimum required lanes" which indicates the minimum number of lanes required to maintain the connectivity of bus routes passing through, and the second term denotes the remaining number of lanes on arc (i, j). The value of k_{1ij} depends on the presence or absence of exclusive bus lanes on the arc; k_{1ij} equals 0 if an exclusive bus lane is allocated, and equals 1 otherwise. Constraints (51)–(52) are added to consider this relation. In fact, they adopt the same concept as in constraints (6)–(7). The revised feasibility constraints for a bimodal link are formulated as follows:

$$k_{1ij} + k_{2ij} + \sum_{p\in\Phi_i, p\neq j} k'_{ip} \geq 1 \tag{46}$$

$$k_{1ji} + k_{2ji} + \sum_{p\in\Phi_i, p\neq j} k'_{pi} \geq 1 \tag{47}$$

$$k_{1ij} + k_{2ij} + \sum_{p\in\Phi_j, p\neq i} k'_{pj} \geq 1 \tag{48}$$

$$k_{1ji} + k_{2ji} + \sum_{p\in\Phi_j, p\neq i} k'_{jp} \geq 1 \tag{49}$$

$$k_{1ij} + k_{2ij} + k_{1ji} + k_{2ji} + \omega_{ij} + \omega_{ji} = k'_{ij} + k'_{ji} + \omega'_{ij} + \omega'_{ji} \tag{50}$$

$$k_{1ij} + \omega_{ij} = 1 \tag{51}$$

$$k_{1ji} + \omega_{ji} = 1 \tag{52}$$

where ω'_{ij} and ω'_{ji} are the current values of the decision variables for exclusive bus lanes, and ω_{ij} and ω_{ji} are the exclusive lane allocation variables. Constraints (51)–(52) indicate the relation between minimum required lanes and exclusive bus lanes. Since $k_{1ij}, k_{1ji}, \omega_{ij}$ and ω_{ji} are binary, one can easily select random binary values satisfying constraints (51) and (52) for these variables. For bimodal links with only two lanes, random lane allocation ends at this point. For other bimodal links, there may be several possible alternatives for k_{2ij} and k_{2ji}. A feasibility interval for k_{2ij} can be defined by using the remaining reduced set of constraints. The lower bound LB_{ij} and the upper bound UB_{ij} of the feasibility interval for k_{2ij} are as follows:

$$LB_{ij} = \max\left\{0, \max\left\{1 - \sum_{p\in\Phi_j, p\neq i} k'_{pj} - k_{1ij},\ 1 - \sum_{p\in\Phi_i, p\neq j} k'_{ip} - k_{1ij}\right\}\right\} \tag{53}$$

$$UB_{ij} = \min\left\{k'_{ij}+k'_{ji}+\omega'_{ij}+\omega'_{ji} - 2, \min\left\{\begin{array}{l} k'_{ij}+k'_{ji}+\omega'_{ij}+\omega'_{ji} - 3 + \sum\limits_{p\in\Phi_j, p\neq i} k'_{jp} + k_{1ji}, \\ k'_{ij}+k'_{ji}+\omega'_{ij}+\omega'_{ji} - 3 + \sum\limits_{p\in\Phi_i, p\neq j} k'_{pi} - k_{1ji} \end{array}\right\}\right\} \tag{54}$$

It suffices to choose a random non-negative integer value for k_{2ij} from the interval, and then to obtain the value of k_{2ji} by solving Eq. (50).

The new solution generation procedure can be summarized as below. The following steps are repeated for each link (i, j) in the solution:

- Randomly decide to change the lane allocation of the link.
- If the link is an "auto link":
  - Define the lane allocation alternatives of the link using Eq. (44).
  - Check the feasibility of defined alternatives using conditions (40)–(43).
  - Choose one of the possible alternatives randomly.
- If the link is a "bimodal link":
  - Choose random binary values for k_{1ij} and k_{1ji}, and then obtain ω_{ij} and ω_{ji} using conditions (51)–(52).
  - Define the lower bound and upper bound for k_{2ij} using conditions (53)–(54).
  - Choose a random value for k_{2ij} from the feasibility interval, and then assign the number of remaining lanes to k_{2ji}.

Any solution resulting in a disconnected network is discarded and the procedure is repeated again. The Pareto-optimal set is updated whenever a random solution is generated.

4.3.4 Evolution mechanism

The evolution process is undertaken by replacing some of the population solutions with the generated offspring solutions with higher fitness values as in Miandoabchi et al. (2010). 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 $|S'| \geq |S_0|$, where S_0 is the set of offspring solutions, then apply SA on a proportion of solutions in S_0, update S_0, and replace $|S_0|$ worst population solutions with all members of S_0.
- If $|S'| < |S_0|$, then apply SA on a proportion of $|S'|$ best offspring solutions and replace $|S'|$ worst population solutions with $|S'|$ best offspring solutions.

4.4 Multi-objective hybrid particle swarm optimization

Particle Swarm Optimization is a nature inspired evolutionary metaheuristic that simulates the social behavior of bird flocking and fish schooling. The term swarm in the algorithm stands for the population of birds or fishes searching for food sources and particle represents each individual bird or fish. Particles explore the environment by following the most successful particle in finding food sources, and by using their own previous experiences. Thus, PSO is a population-based metaheuristic in which the particles (i.e., solutions) in a swarm (i.e., population of solutions) are constantly updated during the solution process. Each element of a particle has a constantly updated velocity value which defines the direction and the extent of change in the value of the

corresponding decision variable (see Kennedy and Eberhart 2001; Clerc 2006). Velocity is the summation of three terms: "inertia" based on the current velocity value of the element, "personal influence" based on the solution element of particle's own best experienced solution so far called local best (*pbest*), and "social influence" based on the solution element of the best particle found in the population during the search process called global best (*gbest*).

PSO was first developed by Kennedy and Eberhart (1995), and since then it has been used in solving various problems. The algorithm was originally designed for problems with continuous variables and thus the continuous version is mostly applied in researches, while the discrete version has recently been under focus in some researches. A continuous version of PSO for the CNDP has been previously developed in Zhang and Gao (2007). In this paper, a multi-objective discrete version of PSO hybridized with SA is proposed, in which SA is applied on each particle after the update phase. The overall procedure of the algorithm is presented below and the details are explained later.

Phase 1: Generate a population of P solutions, compute their fitness values, and build the Pareto-optimal set from the initial population.

Phase 2: Repeat the following procedure for T iterations:

- For each solution in population, repeat the following:
  - Update the solution using the particle update mechanism.
  - Check the connectivity of the solution and discard the solution if the network is disconnected.
  - Check the budget feasibility of the solution and apply the budget reduction sub-routine if needed. If the infeasible solution cannot be repaired, then discard the solution.
  - Update the Pareto-optimal set and compute the fitness value.
  - Apply SA on the updated solution.
  - Decrease the inertia weight, and update *pbest* and *gbest* if required.

The solution encoding, fitness value calculation and initial population generation procedures are the same as those in mHGA. Here, a short version of SA is applied on the updated solution in order to explore the problem space in the vicinity of the generated solution, in the hope of finding better solutions. The embedded SA is similar to the one used in mHGA.

4.4.1 Particle update mechanism

As described earlier, the particle update mechanism consecutively updates each element of the solution until all solution elements are updated. Here the solution elements are the network links (not arcs), in which their update can be characterized by say, changing the number of lanes added to the existing links, changing lane allocation on the existing links, omitting the construction of the new links, and so on. Thus, a network link is updated by changing the values of multiple decision variables rather than a single variable for that link, and therefore a specific velocity type needs to be defined for each variable of the link.

The following types of velocities are defined for each bimodal link l:

V_{0l}: velocity for the lane addition project on link l (if a project exists), and

V_{1ij} and V_{1ji}: velocities for exclusive bus lanes on arcs (i, j) and (j, i), respectively. No velocity is defined for the remaining lanes on arcs, since they depend much on the values of the above mentioned variables.

For auto links, the following types of velocities are defined:

V_{0l}: velocity for the lane addition or link construction project of link l (if a project exists), and
V_{2ij}: velocity for lane allocation on arc (i, j).

A velocity function uses three weights to update each of the velocities stated above: w_0 for the inertia, C_1R for the personal influence, and C_2R for the social influence, where R is a random value between 0 and 1, and C_1 and C_2 are, respectively, the weight parameters for personal and social influence terms. Since the values provided for velocities may not be integers, integer solutions are not guaranteed to obtain. A transformation is performed to convert the resulting non-integer solutions to integer ones by rounding the values to the nearest integer values in the variable's domain.

Various forms of velocity functions have been proposed in the literature, but here the version of Eberhart and Shi (2000) is adopted in which w_0 is slightly decreased in each iteration using a coefficient less than 1 called C_0, in order to control an unacceptable increase of velocities over time. w_0^τ is the value of inertia weight in τth iteration that is equal to $C_0 w_0^{\tau-1}$. Since the start value w_0^1 and the stop value w_0^T are given for the inertia weight, C_0 must be equal to $(w_0^T/w_0^1)^{1/T}$. The procedure of particle update in the τth iteration for an existing link l or a new link l' consisting of arcs (i, j) and (j, i) is as follows:

- Update the link addition variable: If the link is widened, then:
  - Set $V_{0l} = w_0^\tau V_{0l} + C_1R(y_l^{pbest} - y_l) + C_2R(y_l^{gbest} - y_l)$.
  - Set $y_l = y_l + V_{0l}$, and round y_l to the nearest integer in its domain.

  Otherwise if the link is new, then:
  - Set $V_{0l'} = w_0^\tau V_{0l'} + C_1R(u_{l'}^{pbest} - u_{l'}) + C_2R(u_{l'}^{gbest} - u'_{l'})$.
  - Set $u_{l'} = u_{l'} + V_{0l'}$, and round $u_{l'}$ to the nearest binary value.
- Update lane allocation: If the link is bimodal then:
  - Update the allocation of exclusive bus lanes and minimum required lanes on arcs (i, j) and (j, i):
    - Set $V_{1ij} = w_0^\tau V_{1ij} + C_1R(\omega_{ij}^{pbest} - \omega_{ij}) + C_2R(\omega_{ij}^{gbest} - \omega_{ij})$.
    - Set $\omega_{ij} = \omega_{ij} + V_{1ij}$, and round ω_{ij} to the nearest binary value.
    - Set $k'_{1ij} = 1 - \omega_{ij}$.
    - Repeat the above steps for arc (j, i).
  - Define a random allocation for the remaining lanes using the feasibility interval defined by (47) and (48).

    Otherwise, if the link is an existing auto link or a new auto link with $u_{l'} = 1$, then update the allocation of lanes:
  - Set $V_{2ij} = w_0^\tau V_{2ij} + C_1R(k_{ij}^{pbest} - k_{ij}) + C_2R(k_{ij}^{gbest} - k_{ij})$.
  - Set $k_{ij} = k_{ij} + V_{2ij}$, and round k_{ij} to the nearest value in its domain.
  - For an existing link, set $k_{ji} = K_l + 2y_l - k_{ij}$, and for a new link, set $k_{ji} = K_{l'} - k_{ij}$.

4.5 Multi-objective hybrid harmony search

Harmony Search (HS) is a population-based metaheuristic that simulates the music improvising behavior of music composers (especially in jazz). HS has been recently proposed by Geem et al. (2001) as a novel metaheuristic solution procedure. A musician composes a new music piece by gradually adjusting a set of notes until he/she is satisfied by the harmony. The process can be imitated by a problem solution procedure in which the values of the variables (notes) are changed step by step, until the best solution (harmony) in terms of the objective function or fitness value is achieved. Musicians often use three ways to choose and include a note in their new harmony: (1) selecting a note from their memories and previous experiences which is called "Harmony Memory" selection, (2) slightly changing the pitch of the selected note which is called "Pitch Adjustment", and (3) making a random note which is called "Randomization". Harmony memory selection is analogous to choosing the population of solutions; pitch adjustment is analogous to applying a small change in the value of a variable in its neighborhood, and randomization is analogous to selecting a random value for a variable. These three ways of making a new harmony simulates random choices to set values to variables of solutions as will be described later (see Geem 2009 also for the details).

HS as a solution procedure has been recently used in a number of researches, but it has not been used to solve UTNDPs before. In this paper, a new multi-objective hybrid version of HS and SA is developed to solve the proposed problem. SA is applied to improve the solutions produced at each iteration of the main algorithm. The SA used is similar to the one in mHGA. Here is the outline of the multi-objective hybrid version of HS and SA:

Phase 1: Generate a population of P solutions, compute their fitness values, and build the Pareto-optimal set from the initial population.
Phase 2: Repeat the following procedure for T iterations:
- Make a new solution using the new harmony generating mechanism.
- Check the connectivity of the solution and discard the solution if the network is disconnected.
- Check the budget feasibility of the solution and apply the budget reduction sub-routine if needed. If the infeasible solution cannot be repaired, then discard the solution.
- Update the Pareto-optimal set and compute the fitness value.
- Apply SA on the updated solution.
- If the new solution is better than the worst solution in the population, replace the latter with the new solution.

4.5.1 New harmony generating mechanism

The new harmony generating mechanism generates a value for each element of the solution (i.e., a design for each network link) using the three described rules. The value of a variable is chosen from Harmony Memory with the probability R_{accept}, and from a completely random value with the probability $1 - R_{accept}$. Pitch adjustment

is applied to the selected variable with the probability R_{pa}, Pitch adjustment includes applying a small change in the lane allocation of the selected link and is defined as follows:

- Bimodal links with more than two lanes: One of the following decisions is randomly made for one of the randomly chosen arcs of the link:
  - If an exclusive bus lane is present, turn it into a shared auto-bus lane, and otherwise allocate one of the lanes as an exclusive bus lane.
  - Randomly delete one auto lane on one arc, and add one auto lane on the other arc of the same bimodal links.
- Bimodal links with two lanes: For one of the arcs (which is randomly chosen), if an exclusive bus lane is present, turn it into a shared auto-bus lane, and otherwise allocate one of the lanes as an exclusive bus lane.
- Auto links: Randomly change the link into a two-way or one-way link.

The new harmony generation procedure repeats the following steps for each link l or l' of the network to obtain a new solution.

– If $R < R_{accept}$, where R is a random value between 0 and 1,
  - Select the design of link l randomly from one of the population solutions.
  - If $R < R_{pa}$, then apply pitch adjustment on link l or l'.

  Otherwise, apply randomization: randomly select the lane addition or link construction project for link l or l' and then randomly allocate the lanes, taking into account the connectivity of the bus routes in the case of bimodal links.

5 Computational results

5.1 Test instances and data

Because of the novelty of the problem, there is no existing benchmark instance. Hence, a number of used test examples in NDPs, TNDPs, and multi-modal traffic assignment references were selected, and modified to test the performance of the algorithms. Three types of test examples were generated based on three small networks, three medium networks, and one large network.

All the scenario information provided in the references given in Table 2 were used and reasonable values were chosen for the additional required data such as lane addition and link construction projects, the number of lanes in each street, bus lines and their frequencies, walking times for the bus mode, bus passenger capacities, and so on. Construction costs are considered as linear functions of the number of lanes. Table 2 also gives the main characteristics of the test examples. The values in the parentheses are the data provided to the test instances by the authors.

The following values or ranges were used for the required parameters: $\pi = 2$; $\mu = 3$; $\psi = 0.2$ min; $\alpha = 0.15$; $\beta = 4$; $\alpha' = 0.1$; $\beta' = 2$; $\hat{\alpha} = 0.1$; $\hat{\beta} = 2$; $\theta = 0.45\,\text{min}^{-1}$; $\bar{t}_{\hat{i}} = (5, 10)$ min; $n_e = 40$; $f_e = (3, 12)$ buses per hour; $\rho_{ij} = 0.4t^0_{ij}$ monetary unit; $\hat{\rho}_{\hat{i}\hat{j}} = 0.25$ monetary unit; $VOT_t = 0.333$; $VOT_a = 0.333$ and; $VOT_w = 0.667$. The units of all values of times are in monetary units per minute.

Table 2 Testing networks

Network size	Network adopted	Notation no.	Nodes no.	Links no.	OD pairs no.	Bus lines no.	Bimodal lines no.
Small	The Abdulaal and LeBlanc (1979) network	AL	9	12	(16)	(2)	8
	The Nguyen and Dupuis (1984) network used in Kov et al. (2010)	ND	13	19	4	3	12
	A reduced version of Sioux Falls network used in LeBlanc et al. (1975)	SF1	14	19	176	(4)	(17)
Medium	The Nagurney (1984) network	NA1	20	28	8	(5)	(21)
	The basic Sioux Falls network of LeBlanc et al. (1975) used in Abdulaal and LeBlanc (1979)	SF2	24	37	528	5	23
	The Nagurney (1984) network	NA2	25	37	6	(4)	(18)
Large	The Nagurney (1984) network	NA3	40	66	6	(5)	(30)

The monetary unit used in this paper is 1,000 Rials (Currency in Iran). Note that the values of the logit parameters ψ and θ used here are for illustrative purposes and their actual values should be calibrated from the available data. In general, the larger is the value of θ, the smaller is the variance of the perceived generalized travel cost incurred by travelers between the same OD pairs. Moreover, the larger is the value of ψ, the more people travel by bus. To avoid computational complexity, all bus lines are considered as attractive lines and only connectivity between OD pairs is only considered.

5.2 Parameter setting

The parameters of the three algorithms were set by using series of experiments, by searching for parameter ranges in similar algorithms from related papers to find some initial ranges, by considering the number of solutions generated from the three algorithms, and by adopting the used parameter settings in the previous work of the authors (Miandoabchi et al. 2010). Table 3 depicts the parameter settings.

The maximum number of the solutions generated by each algorithm which indicates the approximate total computational effort can be used as a guideline for setting the parameters of the algorithms. For mHGA, the number can be shown as $G \times (2/3 \times N \times itr + 2 \times N)$, where N is the number of nodes and itr the number of solutions generated by SA, since $2 \times N$ solutions are generated using the crossover operator, and itr solutions are generated in the SA iterations from each of 1/3 of the $2 \times N$ solutions. The proportion 1/3 is chosen to deal with the trade-off between high computational effort

Table 3 Parameter settings for the algorithms

Algorithm	Main parameters	SA parameters
mHGA	$P = 60$	Start temperature: 5
	$G = 300$	Reduction rate for small problems: 0.9
	$k = 1/3$ of the selected offspring solutions to be inserted in the population	Reduction rate for medium and large problems: 0.8 Stop temperature: 1
mHPSO	$P = 30$	Start temperature: 5
	T for small problems $= 8 \times$ N	Reduction rate: 0.9
	T for medium and large problems $= 4 \times$ N	Stop temperature: 1
	$w_o =$ start at 0.95 and stop in 0.2	
	$C_1 = C_2 = 1.25$	
mHHS	$P = 50$	Start temperature: 5
	T for small problems $= 225 \times$ N	Reduction rate: 0.9
	T for medium and large problems $= 125 \times$ N	Stop temperature: 1
	$R_{accept} = 0.9$	
	$R_{pa} = 0.3$	

and the solution quality. For mHPSO, the number can be shown as $P \times T \times (itr + 1)$, since for each of the T iterations, each of P solutions is updated and evaluated once, and then *itr* SA iterations is applied on each of the solutions. Finally, for mHHS, the number is $T \times (itr + 1)$ since for each of the T iterations a solution is generated, evaluated, and then SA is applied on the solution.

The parameter values for mHGA were chosen to be as the values used in Miandoabchi et al. (2010). In order to make the algorithms results comparable to each other for each test example, the number of solution evaluations must be equal for all algorithms. This requires that the parameters of the other two algorithms (i.e., mHPSO and mHHS) must be changed with the size of the problem. Since the general structure of the three algorithms are somewhat similar to each other, we chose to vary the number of iterations T based on the problem size, and keep the other parameter values fixed. To do this, at first, experimental tests were carried out for mHPSO and mHHS using the test example ND for various combinations of the parameters. Then, by keeping all the parameters' values fixed, the number of iterations T was made dependent on N to keep the total number of solution evaluations equal to that of mHGA for each test example.

Again, for the reason of the similarity of the algorithm structures, the SA parameters of mHGA for the small test example ND were used for both mHPSO and mHHS. In addition, the SA parameters for small problems were chosen for all problem sizes. The SA parameters for mHGA are different for small and medium-large text examples because the computational effort for mHGA which depends on N, grows rapidly with the increase of the problem size, and makes the runtimes significantly high. Thus, different SA parameter settings were set to deal with the high runtimes.

Finally, the starting parameter of the distance-based fitness evaluation d_1 was set to 2 after performing experimental runs with a fixed set of algorithm parameters for mHHS.

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 2.2 GHz CPU, and a 2 GB RAM. Each algorithm was run 5 times.

5.4 Performance comparison

Since the algorithms are multi-objective ones, special effectiveness measures are required to assess the quality of the solutions obtained. Meanwhile, the algorithms can be compared by investigating the best values of objective functions they have obtained in Pareto-optimal solutions. This can help to identify the extent of the ideal Pareto-optimal solution space that has been explored by the algorithms.

Here, two types of measures are used to investigate the quality of the Pareto-optimal sets generated by the algorithms. The first measure M_1 gives the size of the Pareto-optimal set and the second measure M_2 is defined to identify how the algorithm performs in overall, in comparison to the other algorithms. M_2 is a new measure proposed in Miandoabchi et al. (2010) which is adopted in this paper. It is based on a set coverage measure proposed in Zitzler et al. (2000), which is used for a pair-wise comparison of algorithms in terms of the fraction of Pareto-optimal solutions obtained by one algorithm that dominates the solutions obtained from another algorithm. The measure M_2 is defined as follows:

$$M_2(X_i) = \sum_{j \neq i} C(X_i, X_j) - \sum_{j \neq i} C(X_j, X_i) \tag{55}$$

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

$$C(X_i, X_j) = \frac{\left|\left\{a_j \in X_i; \exists a_i \succ= a_j\right\}\right|}{\left|X_j\right|} \tag{56}$$

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 calculates the fraction of solutions in set X_j is dominated by or equal to at least one solution in X_i.

The measure M_2 indicates the overall strength of an algorithm with respect to other algorithms, in terms of the quality of the Pareto-optimal set (Miandoabchi et al. 2010). The idea is to sum up all C values that indicate the fraction of solutions which dominate the solutions of other algorithms, and to subtract all C values that indicate the fraction of solutions which are dominated by solutions of other algorithms. In other words, fractions of non-dominated solutions of the algorithm contribute to increase the

Table 4 Summary of computational results

Problem code	mHGA			mHPSO			mHHS		
	M_1	M_2	Runtime[a]	M_1	M_2	Runtime[a]	M_1	M_2	Runtime[a]
AL	2	2.00	701	3	−1.16	888	4	−0.84	789
ND	33	−1.82	765	28	1.44	524	19	0.38	277
SF1	15	−1.94	1,958	5	1.81	1,970	6	0.13	889
NA1	57	−1.71	6,131	25	1.77	1,314	19	−0.06	953
SF2	35	0.13	11,004	12	1.48	3,485	7	−1.61	1,784
NA2	45	−1.20	10,262	15	0.88	1,740	11	0.33	1,656
NA3	28	−0.45	40,071	10	0.93	11,153	10	−0.48	9,104

[a] Average runtime (in seconds)

measure value and fractions of non-dominated solutions of the algorithm contribute to decrease the measure value.

The summary of the average values of the measures and the average runtimes are given in Table 4. Regarding to the values of the measure M_1 in Table 4, mHGA has 6 highest values. The second ranking belongs to mHPSO generated 5 second highest values, and finally mHHS has 1 highest and 2 second highest values. With respect to the values of the measure M_2 for the algorithms, mHPSO obtains 6 highest values. mHHS has the second ranking obtaining 6 second highest values, and mHGA gains the third ranking producing 1 highest and 2 second highest values. It can be seen that mHGA outperforms both algorithms in terms of the size of the Pareto-optimal set and mHPSO performs the best in terms of the quality of the Pareto-optimal set.

To evaluate the capability of the algorithms in achieving the Pareto optimal solutions, a branch and bound (B&B) method was applied to solve the problem. The particular branch and bound method used here deals with multiple objectives rather than a single one, and it does not apply lower bound calculation. Each time a feasible solution is constructed, it is checked to be added to the set of Pareto-optimal set and thus the output of the method is a set of Pareto-optimal solutions. The method was solved only for the smallest example AL, since the run time of the method is very high even for the smallest network. The method was stopped after running for more than 40 h, and the obtained 9 members in the Pareto-optimal set, which was considered as the best solution set found so far. Then the union of Pareto-optimal sets of all 5 runs was obtained for each algorithm, and the pair-wise comparison was performed between the method and the three algorithms using measure (56). The results of the comparison are given in Table 5. It can be observed that the three algorithms dominate all the solutions of the branch and bound method.

Another comparison can be made for the algorithms runtimes. mHHS has the first rank with 6 lowest and 1 second lowest runtimes. mHPSO has the second ranking with 5 second lowest runtimes. mHGA have 1 lowest and 1 second lowest runtimes.

Figure 4 illustrates the average runtimes of the algorithms. Although the maximum number of solution evaluations was set to be equal for the algorithms in all test examples, it is observed that the runtime of mHGA rapidly grows with the increase

Table 5 Pair-wise comparison between B&B and the algorithms

Algorithm	$C(X_*, X_{\text{B\&B}})$	$C(X_{\text{B\&B}}, X_*)$
mHGA	1.00	0
mHPSO	1.00	0
mHHS	1.00	0

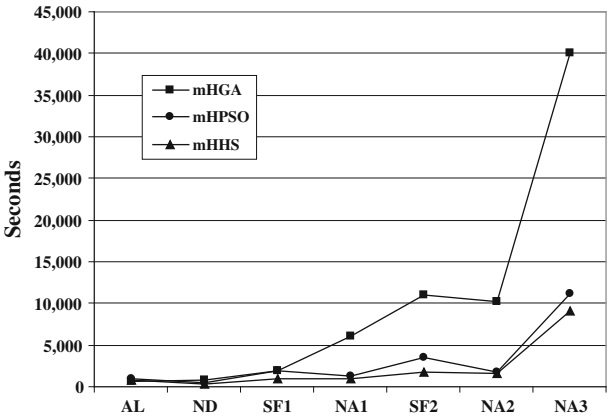

Fig. 4 Average runtimes comparison

Table 6 The best objective function values

Example	mHGA		mHPSO		mHHS	
	Z_1	Z_2	Z_2	Z_1	Z_2	Z_2
AL	139,400	0.49897	96,088	0.35536	98,308	0.35608
ND	451,310	0.78329	916,740	0.77241	886,490	0.77077
SF1	427,400	0.67800	461,710	0.74095	452,950	0.69701
NA1	462,480	0.88442	768,920	0.90268	638,570	0.86908
SF2	14,010,000	0.62190	14,788,000	0.64528	7,671,500	0.60222
NA2	464,660	0.84487	686,220	0.83849	623,920	0.82531
NA3	199,992	0.82971	355,724	0.69828	324,288	0.65582

of the problem size. This may be explained by the fact that the numbers of infeasible solution generated in terms of network disconnectedness (not the budget feasibility) in mHPSO and mHHS are higher than that of mHGA, since mHGA can obtain more feasible solutions by using the crossover operator depicted in Sect. 4.3.

A performance comparison can be made by assessing the best values for each of the objective functions in the Pareto-optimal set obtained by each algorithm. Table 6 depicts the best objective function values obtained. The values presented in the table are the best values among all 5 runs.

A comparison of values of the two objective functions in all 7 examples (i.e., 14 values of objective functions) for each algorithm reveals that mHPSO obtained 9 best

and 3 second best values; mHGA produced 5 best and 3 second best values, and mHHS generated 8 second best values. To illustrate their relative performance more clearly, a comparison was performed by ranking the algorithms for each test example in terms of the values of Z_1 and Z_2, and then the total number of times that each algorithm gained the first or second ranking was counted.

A quick summary for the algorithms' ranking under each measure is shown as follows:

- M_1: mHGA>mHPSO>mHHS
- M_2: mHPSO>mHHS>mHGA
- Best Pareto-optimal objective values: mHPSO>mHGA>mHHS
- Runtime: mHHS>mHPSO>mHGA

The rankings suggest that mHPSO outperforms others in terms of M_2 and the best values for individual objective functions. mHGA outperforms others in terms of the size of the solution set, and finally mHHS has the lowest runtime. Overall, mHPSO can be considered as the most desirable solution procedure for this problem, since it is capable of obtaining much higher solution set quality M_2 with a much lower runtime compared to mHGA for medium and large networks (see Fig. 4), and mHPSO can obtain higher solution set quality in terms of M_1 and M_2 with a similar computation time compared to mHHS.

6 Conclusions and future research

In this paper, a bi-objective bimodal network design problem including the automobile and bus modes is investigated. Five types of decisions are included the problem: constructing new streets, adding lanes to the existing streets where it is possible, determining the directions of one-way streets, determining lane allocation in two-way streets with bus routes, and finally determining where to allocate exclusive bus lanes and where to use street lanes for shared use of automobiles and buses. The main contribution of the paper lies on proposing a new road network design problem with the inclusion of the decisions for bus network, by taking into account the restrictions of the street lane allocation on the existing bus routes, the effects of the automobile and bus flows on each other, and the elasticity of the travel demands. The problem is modeled as a mathematical problem with equilibrium constraints, which is intrinsically a bi-level problem. The upper level problem corresponds to the decision making for the main decision variables, and the lower level problem corresponds to the modal-split/assignment problem. Because of the intrinsic complexity and non-convexity of the model, three hybrid multi-objective population-based metaheuristic algorithms are developed to solve for good rather than exact solutions. The proposed algorithms are the hybrid genetic algorithm, particle swarm optimization, and harmony search in which they incorporate simulated annealing. These algorithms use a distance-based fitness evaluation method to capture the multi-objectivity of the problem and generate a set of Pareto-optimal solutions.

The algorithms are evaluated by solving test networks taken from the literature and customizing them for the problem on hand, and their performances are compared to each other using various performance measures. The computational tests indicate that

the proposed multi-objective particle swarm optimization algorithm outperforms the other two algorithms in terms of the quality of the Pareto-optimal sets with reasonable runtimes.

This paper opens a new line of research about the simultaneous design of urban road and transit networks. Various forms of problems can be considered by combining different RNDPs and bus network design problems. For example, one extension to the problem which is currently under study by the authors can be including the redesign of the bus routes in RNDPs. The street directions and network expansion projects can be determined by concurrently redesigning or modifying the routes and frequencies of existing bus lines, without changing their terminal nodes. Another research subject is to combine the design of the bus network including bus routes, terminals and frequencies with the decisions of RNDPs. Besides, the current formulation cannot be applied to some historical urban centers in Europe and some densely populated cities in Asia, where there is no way to ensure that both forth and back routes of bus lines pass through the same link (e.g., the one-way one-lane street has high bus demand but there is no extra space for widening the street). In the future, the requirement of passing through the same link for both forth and back routes can be relaxed to handle this case. Finally, including signal timings and travel times for crossing intersections lead to a more realistic but a higher complexity network design problem to solve.

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

References

Abdulaal M, LeBlanc LJ (1979) Methods for combining modal split and equilibrium assignment models. Transp Sci 13(4):292–314

Bellei G, Gentile G, Papola N (2002) Network pricing optimization in multi-user and multimodal context with elastic demand. Transp Res B 36(9):779–798

Ben-Ayed O, Boyce DE, Blair CE (1988) A general bilevel linear programming formulation of the network design problem. Transp Res B 22(4):311–318

Cantarella GE, Vitetta A (2006) The multi-criteria road network design problem in an urban area. Transportation 33(6):567–588

Cantarella GE, Pavone G, Vitetta A (2006) Heuristics for urban road network design: lane layout and signal settings. Eur J Oper Res 175(3):1682–1695

Chen A, Yang C (2004) Stochastic transportation network design problem with spatial equity constraint. Transp Res Rec 1882:97–104

Chen A, Yang H, Lo HK, Tang WH (2002) Capacity reliability of a road network: an assessment methodology and numerical results. Transp Res B 36(3):225–252

Chen A, Subprasom K, Ji Z (2006) A simulation-based multi-objective genetic algorithm (SMOGA) for build-operate-transfer network design problem. Optim Eng J 7(3):225–247

Chen A, Kim J, Zhou Z, Chootinan P (2007) Alpha reliable network design problem. Transp Res Rec 2029:49–57

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

Chen A, Kim J, Lee S, Kim Y (2010b) Stochastic multi-objective models for network design problem. Expert Syst Appl Int J 37(2):1608–1619
Chiou SW (2008a) A hybrid approach for optimal design of signalized road Network. Appl Math Model 32(2):195–207
Chiou SW (2008b) A non-smooth model for signalized road network design problems. Appl Math Model 32(7):1179–1190
Chiou SW (2009a) A subgradient optimization model for continuous road network design problem. Appl Math Model 33(3):1386–1396
Chiou SW (2009b) Simultaneously optimizing link tolls and signal settings in a road network. Appl Math Model 33(5):2311–2323
Chiou SW (2009c) An efficient algorithm for optimal design of area traffic control with network flows. Appl Math Model 33(6):2710–2722
Clegg J, Smith M, Xiang Y, Yarrow R (2001) Bilevel programming applied to optimising urban transportation. Transp Res B 35(1):41–70
Clerc M (2006) Particle swarm optimization. ISTE Ltd, UK
D'Acierno L, Gallo M, Montella B (2005) A multimodal approach for managing transportation design problems of real size networks. In: Jaszkiewicz A, Kaczmarek M, Zak J, Kubiak M (eds) Advanced OR and AI methods in transportation. Publishing House of Poznan University of Technology, Poznan pp 200–205
D'Acierno L, Gallo M, Montella B (2006) Optimisation models for the urban parking pricing problem. Transp Policy 13(1):34–48
Dafermos S (1972) The traffic assignment problem for multiclass-user transportation networks. Transp Sci 6(1):73–87
De Cea J, Fernández E (1993) Transit assignment for congested public transport systems: an equilibrium model. Transp Sci 27(2):133–147
Desaulniers G, Hickman MD (2007) Public transit. In: Barnhart C, Laporte G (eds) Handbook in operations research and management science transportation, vol 14. Elsevier, Amsterdam pp 69–127
Drezner Z, Salhi S (2000) Selecting a good configuration of one-way and two-way routes using tabu search. Control Cybern 29(3):725–740
Drezner Z, Salhi S (2002) Using hybrid metaheuristics for the one-way and two-way network design problem. Naval Res Logist 49(5):449–463
Drezner Z, Wesolowsky GO (1997) Selecting an optimum configuration of one-way and two-way routes. Transp Sci 31(4):386–394
Drezner Z, Wesolowsky GO (2003) Network design: selection and design of links and facility location. Transp Res A 37(3):241–256
Eberhart RC, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 congress on evolutionary computation, vol 1, pp 84–88
Elshafei EH (2006) Decision-making for roadway lane designation among variable modes. PhD Thesis, University of Maryland, USA
Ferrari P (1999) A model of urban transport management. Transp Res B 33(1):43–61
Friesz TL (1985) Transportation network equilibrium, design and aggregation: key developments and research opportunities. Transp Res A 19(5–6):413–427
Gallo M, D'Acierno L, Montella B (2010) A meta-heuristic approach for solving the urban network design problem. Eur J Oper Res 201(1):144–157
Gao Z, Song Y (2002) A reserve capacity model of optimal signal control with user-equilibrium route choice. Transp Res B 36(4):313–323
Gao Z, Sun H, Shan LL (2004) A continuous equilibrium network design model and algorithm for transit systems. Transp Res B 38(3):235–250
Gao Z, Wu J, Sun H (2005) Solution algorithm for the bi-level discrete network design problem. Transp Res B 39(6):479–495
Geem ZW (2009) Music-inspired harmony search algorithm: theory and applications. Springer, Berlin
Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
Guihaire V, Hao JK (2008) Transit network design and scheduling: a global review. Transp Res A 42(10):1251–1273
Hamdouch Y, Florian M, Hearn DW, Lawphongpanich S (2007) Congestion pricing for multi-modal transportation systems. Trans Res B 41(3):275–291

Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Michigan
Huang HJ (2002) Pricing and logit-based mode choice models of a transit and highway system with elastic demand. Eur J Oper Res 140(3):562–570
Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol 4, pp 1942–1948
Kennedy J, Eberhart RC (2001) Swarm intelligence. Morgan Kaufmann, USA
Kov M, Fukuda D, Yai T (2010) Frequency design of bus transit mixed urban transport network. Asian Transp Stud 1(4):352–367
LeBlanc LJ (1975) An algorithm for the discrete network design problem. Transp Sci 9(3):183–199
LeBlanc LJ, Morlok EK, Pierskalla WP (1975) An efficient approach to solving the road network equilibrium traffic assignment problem. Transp Res 9:309–318
Lee CK, Yang KI (1994) Network design of one-way streets with simulated annealing. Pap Reg Sci 73(2):119–134
Li SG, Ju YF (2009) Evaluation of bus-exclusive lanes. IEEE Trans Intell Transp Syst 10(2):236–245
Lo HK, Yip CW, Wan KH (2003) Modelling transfer and non-linear fare structure in multi-modal network. Transp Res B 37(2):149–170
Magnanti TL, Wong RT (1984) Network design and transportation planning: models and algorithms. Transp Sci 18(1):1–55
Matis P, Kohání M (2010) Very large street routing problem with mixed transportation mode. Cent Eur J Oper Res doi:10.1007/s10100-010-0159-2
Miandoabchi E, Zanjirani Farahani R, Szeto WY (2010) Multi-objective discrete urban road network design. Comput Aided Civil Infrastructure Eng (submitted)
Migdalas A (1995) Bilevel programming in traffic planning: models, methods, and challenge. J Glob Optim 7(4):381–405
Meng Q, Khoo HL (2008) Optimizing contraflow scheduling problem: model and algorithm. J Intell Transp Syst 12(3):126–138
Meng Q, Yang H (2002) Benefit distribution and equity in road network design. Transp Res B 36(11):19–35
Meng Q, Yang H, Bell MGH (2001) An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. Transp Res B 35(1):83–105
Meng Q, Ling H, Cheu RL (2008) Microscopic traffic simulation model-based optimization approach for the contraflow lane configuration problem. J Transp Eng ASCE 134(1):41–49
Mesbah M, Sarvi M, Currie G (2008) New methodology for optimizing transit priority at the network level. Transp Res Rec 2089:93–100
Nagurney A (1984) Comparative tests of multimodal traffic equilibrium methods. Transp Res B 18(6):469–485
Nguyen S, Dupuis C (1984) An efficient method for computing traffic equilibria in networks with asymmetric transportation costs. Transp Sci 18(2):185–202
Nguyen S, Pallottino S, Gendreau M (1998) Implicit enumeration of hyperpaths in a logit model for transit networks. Transp Sci 32(1):54–64
Nuzzolo A, Russo F, Crisalli U (2001) A doubly dynamic schedule-based assignment model for transit networks. Transp Sci 35(3):268–285
Osyczka A, Kundu S (1995) A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. Struct and Multidiscip Optim 10(2):94–99
Poorzahedy H, Abulghasemi F (2005) Application of ant system to network design problem. Transportation 32(3):251–273
Poorzahedy H, Rouhani OM (2007) Hybrid meta-heuristic algorithms for solving network design problem. Eur J Oper Res 182(2):578–596
Poorzahedy H, Turnquist MA (1982) Approximate algorithms for the discrete network design problem. Transp Res B 16(1):45–55
Reimann M, Ulrich H (2006) Comparing backhauling strategies in vehicle routing using ant colony optimization. Cent J Oper Res 14(2):105–123
Roberts FS, Xu Y (1988) On the optimal strongly connected orientations of city street graphs I: large grids. SIAM J Discret Math 1(2):199–222
Roberts FS, 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 FS, 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 FS, Xu Y (1994) On the optimal strongly connected orientations of city street graphs: IV. Four east–west avenues or north–south streets. Discret Appl Math 49(1–3):331–356
Seo YU, Park JH, Jang H, Lee YI (2005) A study on setting-up methodology and criterion of exclusive bus lane in urban area. Proc East Asia Soc Transp Stud 5:339–351
Sheffi Y (1985) Urban transportation networks: equilibrium analysis with mathematical programming methods. Prentice-Hall, Englewood Cliffs
Sörensen K (2006) Route stability in vehicle routing decisions: a bi-objective approach using metaheuristics. Cent Eur J Oper Res 14(2):193–207
Steenbrink PA (1974) Transport network optimization in the Dutch integral transportation study. Transp Res 8(1):11–27
Szeto WY, Lo HK (2005) Strategies for road network design over time: robustness under uncertainty. Transportmetrica 1(1):47–63
Szeto WY, Lo HK (2006) Transportation network improvement and tolling strategies: the issue of intergeneration equity. Transp Res A 40(3):227–243
Szeto WY, Lo HK (2008) Time-dependent transport network improvement and tolling strategies. Transp Res A 42(2):376–391
Szeto WY, Jaber XQ, O'Mahony M (2010) Time-dependent discrete network design frameworks considering land use. Comput Aided Civil Infrastructure Eng 25(6):411–426
Szeto WY, Solayappan M, Jiang Y (2011) Reliability-based transit assignment for congested stochastic transit networks. Comput Aided Civil Infrastructure Eng (in press)
Tehran Times (2009) Tehran's historic street goes one-way northbound. http://www.tehrantimes.com/index_View.asp?code=201298
Tehran Times (2010) BRT line connects south Tehran to the north. http://tehrantimes.com/PDF/10923/10923-7.pdf
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. Netw Sp Econ 7(3):213–240
Van Nes R (2002) Design of multimodal transport networks. PhD Thesis, Delft University, Netherlands
Wong SC, Yang H (1997) Reserve capacity of a signal-controlled road network. Transp Res B 31(5):397–402
Yang H, Bell MGH (1998) Models and algorithms for road network design: a review and some new developments. Trans Rev 18(3):257–278
Yin Y (2000) Genetic-algorithm-based approach for bi-level programming models. J Transp Eng ASCE 126(2):115–120
Yin Y (2002) Multiobjective bi-level optimization for transportation planning and management problems. J Adv Transp 36(1):93–105
Yin Y, Madanat S, Lu X (2009) Robust improvement schemes for road networks under demand uncertainty. Eur J Oper Res 198(2):470–479
Ying JQ, Yang H (2005) Sensitivity analysis of stochastic user equilibrium flows in a bi-modal network with application to optimal pricing. Transp Res B 39(9):769–795
Zhang H, Gao Z (2007) Two-way road network design problem with variable lanes. J Syst Sci Syst Eng 16(1):50–61
Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195