Multi-objective Discrete Urban Road Network Design

Elnaz Miandoabchi [1], Farzaneh Daneshzand [2], Reza Zanjirani Farahani [3*], W.Y. Szeto [4]

(1) Logistics and Supply Chain Management Research Group, Institute for Trade Studies and Research, Tehran, Iran
(2) Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran
(3) Kingston Business School, Kingston University, Kingston Hill, Kingston Upon Thames, Surrey KT2 7LB, The UK
(4) Department of Civil Engineering, The University of Hong Kong, Hong Kong

Abstract

This paper addresses the problem of designing urban road networks in a multi-objective decision making framework. Given a base network with only two-way links, and the candidate lane addition and link construction projects, the problem is to find the optimal combination of one-way and two-way links, the optimal selection of network capacity expansion projects, and the optimal lane allocations on two-way links to optimize the reserve capacity of the network, and two new travel time related performance measures. The problem is considered in two variations; in the first scenario, two-way links may have different numbers of lanes in each direction and in the second scenario, two-way links must have equal number of lanes in each direction. The proposed variations are formulated as mixed-integer programming problems with equilibrium constraints. A hybrid genetic algorithm, an evolutionary simulated annealing, and a hybrid artificial bee colony algorithm are proposed to solve these two new problems. A new measure is also proposed to evaluate the effectiveness of the three algorithms. Computational results for both problems are presented.

Keywords: Urban road; Network design; Transportation; Multi-objective; Hybrid metaheuristics.

1. Introduction

Continuous growth of travel demand in urban transportation networks is one the most challenging and important issues which decision makers often face with. Some engineering solutions to deal with this issue are: road expansion, road construction, changing the configuration of transportation networks, modification of one-way street orientations, revising toll, signal, and parking charge settings, and so on. These engineering solutions can be determined either in an ad hoc manner or by formulating and then solving the Network Design Problem.

The Network Design Problem, or NDP in short, can be defined as the problem of determining the optimal engineering solutions to improve the transportation network (e.g., Chen and Yang, 2004; Szeto and Lo, 2005, 2006, 2008; Chen *et al.*, 2006, 2007; Teodorović and Edara, 2007; Dahl and Minken, 2008; Pacheco *et al.*, 2009; Lo and Szeto, 2009; Szeto *et al.*, 2010; Szeto and Wu, 2011). For the detailed reviews of NDP, the readers can refer to Steenbrink (1974a), Magnanti and Wong (1984), Friesz (1985), Migdalas (1995), and Yang and Bell (1998a).

* Corresponding. Fax: +44 (0)20 8417 7026; Tel: +44 (0)20 8417 5098; Email: zanjiranireza@gmail.com

The NDP model has been shown to have a bi-level structure (Yang and Bell, 1998a). In this type of bi-level optimization problems, the network authority declares its strategy and the users respond to the strategy. Historically, bi-level optimization problems were investigated in the field of game theory. Game theory provides a framework to determine the optimal decisions of the players. The framework can be regarded as a set of coupled optimization problems. Wardrop (1952) was the first one to use the concept of game theory to traffic assignment problems. In fact, the variational inequality formulation of the user equilibrium traffic assignment was the result of the finding that the conditions of equilibrium for the Nash N-person non-cooperative game are the same as those of a zero sum two person game. This kind of game can be modeled by a variational inequality formulation (Fisk, 1984). Therefore, having a close correspondence with the concept of Stackelberg or leader-follower game, this problem is formulated as a bi-level program which can be reduced into a single-level program by expressing the lower level problem as constraints. Many authors have discussed different solution methods of NDP from the view point of Stackelberg and Nash equilibrium conditions (Fisk, 1984; Friesz and Harker, 1985; Chen and Ben-Akiva, 1998; Yang et al., 1994; Yang, 1995) and many others used these results in their network design problems.

The NDP can be categorized in terms of strategic or long-term decisions (such as street expansions or constructions), tactical or medium-term issues (like street orientations), and operational or short-term issues (like signal setting) (Magnanti and Wong, 1984). Based on the decisions made on network topology and network parameters, the NDP can also be categorized in three types, namely the Continuous Network Design Problem (CNDP), the Discrete Network Design Problem (DNDP), and the Mixed Network Design Problem (MNDP).

The input of the CNDP is the network topology, and the output is the optimal values of continuous decision variables or network parameters, such as signal setting, toll setting, and capacity expansion. Some studies related to the CNDP are: Steenbrink (1974a), Abdulaal and LeBlanc (1979), Dantzig *et al.* (1979), Meng *et al.* (2001), and Meng and Yang (2002), and Teklu *et al*. (2007).

The DNDP, in contrast, is concerned with just discrete decision variables like street orientation or street construction. This type of problem has been investigated by Steenbrink (1974b), LeBlanc (1975), Poorzahedy and Turnquist (1982), Lee and Yang (1994), Drezner and Wesolowsky (1997; 2003), Drezner and Salhi (2000; 2002), Gao *et al.* (2005), Poorzahedy and Abulghasemi (2005), Poorzahedy and Rouhani (2007), Meng and Khoo (2008), and Meng *et al.* (2008).

The MNDP or mixed network design problem, as can be implied by its name, involves both discrete and continuous network design variables. It deals with both network topology and network parameters. Compared with the CNDP and the DNDP, the MNDP has received less attention (e.g., Cantarella and Vitetta, 2006; Cantarella *et al.*, 2006; Gallo *et al.*, 2010; Luathep *et al.*, 2011).

Concerning the solution methods, it should be noticed that the combinatorial nature of the DNDP and MNDP makes them much harder to solve than the CNDP. The exact method such as branch and bound algorithms can only deal with small networks which are not realistic enough. Therefore, meta-heuristics and their hybrids have been commonly developed or applied to solve the DNDP and MNDP with large, realistic networks.

There are a number of observations that can be drawn from the existing NDP literature, which are the base of motivation for investigating the problem introduced in this paper. These observations are concerned with the type and number of objective functions, decision variables and their combinations.

First, among the above NDP studies, we observe that total travel time is a widely used objective function but reserve capacity is not. Reserve capacity is defined as the largest multiplier applied to a given, existing demand matrix without violating the street flow capacities, and has been previously investigated for network intersections by Webster and Cobbe (1966), Allsop (1972), and Wong (1996). The reserve capacity as an objective function of NDP was first suggested by Yang and Bell (1998a) but has only been adopted in a few CNDP papers such as Ziyou and Yifan (2002) and Yang and Wang (2002). The reserve capacity concept has not been considered in MNDP studies, and in only one DNDP study (Miandoabchi and Farahani, 2011).

Indeed, minimizing the total travel time or cost is not equivalent to maximizing reserve capacity (Yang and Wang, 2002). Moreover, according to Yang and Bell (1998b), there are at least two advantages of adopting reserve capacity as an objective function. First, the occurrence of capacity paradox can be avoided when reserve capacity is considered in network design. Second, it allows us to predict how much additional demand can be accommodated by the road network after improvement, and hence other related policies for traffic restraint and growth can be established. Third, the upper-level objective function takes a simple linear form and hence the problem is much easier to solve than the NDP with other non-linear objective functions. Fourth, the optimal decision based on reserve capacity is not sensitive to the demand level. This is desirable when uncertainty exists in traffic demands. All these facts seem to indicate that reserve capacity should be considered in DNDPs and MNDPs.

Second, we observe that most NDP papers deal with just one objective function for simplicity. Only few studies have considered multiple objectives for the NDP (e.g., Friesz *et al.*, 1993; Cantarella and Vitetta, 2006; Cantarella *et al.*, 2006; Russo and Vitetta, 2006; Wang and Yang, 2002). But as known, both the network authority and the network users in urban transportation areas are concerned with a wide range of evaluation criteria and the problem is multi-objective in nature. The only multi-objective study with reserve capacity is done with Wang and Yang (2002), which consider reserve capacity and total travel time as the two objectives.

The third observation suggests that apart from using single strategic, tactical or operational decisions in discrete network design, their combination has been studied in a number of studies. For example, the following types of combinations of strategic and tactical decisions can be distinguished in the literature: (a) street expansion and new street construction (e.g. Poorzahedy and Turnquist, 1982; Poorzahedy and Abulghasemi, 2005; Poorzahedy and Rouhani, 2007; Yang and Bell, 1998a; Zhang and Gao, 2009), (b) street orientation and lane allocation (e.g. Cantarella and Vitetta, 2006), (c) new street construction and street orientation (e.g. Drezner and Wesolowsky, 2003), and (d) street expansion with street orientation and lane allocation (e.g. Miandoabchi and Farahani, 2011).

In regard to reserve capacity optimization for DNDPs, Miandoabchi and Farahani (2011) is the only existing study in the literature that considers various decisions to improve reserve capacity. While the construction of new streets can increase the capacity of the network like other strategic and tactical decisions, this improvement strategy has not been used for improving the reserve capacity. Thus, this strategy can be considered together with other decisions simultaneously to further improve the reserve capacity.

Following the above considerations, in this paper we study the discrete network design problem with multiple objective functions, having the reserve capacity maximization as one of the objective functions. Explicitly, the following decisions are investigated:

(a) Adding lanes to the existing network streets,

(b) Constructing new streets,

(c) Determining the orientation of one-way streets, and

(d) Lane allocations in two-way streets.

Although the first two decisions are strategic and the last two are tactical in the NDP context, simultaneously considering all of them can lead to better results in terms of the network performance. Modifying two-way streets to one-way streets can result in improved network performance with fewer requirements for street construction and expansion projects. This can be very advantageous, since such projects require high levels of budget and also require the utilization of land that is expensive and valuable in urban areas.

The proposed problem has three objective functions; reserve capacity and two new travel related objective functions. These objective functions are included, as the maximization of reserve capacity cannot always minimize travel time related measures (Yang and Wang, 2002), which are concerned by network users and network authority.

We define two variations of the DNDP: one restricts two-way links in which there is the same number of lanes in each direction and the other one does not, i.e. the number of lanes in one direction can be different from that of the opposing direction. The problems are formulated as bi-level mathematical programs with equilibrium constraints (MPEC), in which each problem model is bi-level, Stackelberg model with two lower level problems. The inherent complexity of the proposed problem variations and their non-convex nature suggests the application of meta-heuristic solution procedures. Since the problems are new and there are many constraints, directly applying existing metaheuristic to solving them is not possible. In this regard, we developed three new multi-objective hybrid evolutionary algorithms. All algorithms generate as set of Pareto-optimal solutions. To summarize, the following are the contributions of the paper in the literature:

1. Using the combination of street construction with other strategic and tactical decisions to improve the reserve capacity,
2. Adapting a unique combination of triple objective functions in the DNDP,
3. Proposing two new travel-time related objective functions, and
4. Developing three improved meta-heuristics for the two new problems.

Table 1 compares the proposed problem in this paper with the problems examined in the existing NDP studies. In this table, only studies with strategic or tactical decision variables are included.

Table 1. Comparison of the proposed problems with the other studies

Reference		This paper	Miandoabchi and Farahani (2011)	Miandoabchi *et al.* (2012)	Cantarella and Vitetta (2006)	Ziyou and Yifan (2003)	Wang and Yang (2002)
NDP type		DNDP	DNDP	Bi-modal DNDP	MNDP	CNDP	CNDP
Decisions	Street Orientation	✓	✓	✓	✓	-	-
	Lane Allocation	✓	✓	✓	✓	-	-
	Street Expansion	✓	✓	✓	-	✓	✓
	Street Construction	✓	-	✓	-	-	-
	Other	-	-	Bus lane allocations	Signal setting; parking space	Signal setting	-
Objective Functions	Total Travel Time	-	-	-	✓	-	✓
	Reserve Capacity	✓	✓	-	-	✓	✓
	Other	Two travel time related	-	Total user benefit; bus demand share	Bus, pedestrian and people related functions	-	-

The rest of the paper is organized as follows. In the next section, the proposed problem is defined precisely. In section 3, the notations and mathematical formulations of the two problem variations are presented. Section 4 describes the solution method, i.e. the meta-heuristic solution procedures, in details. Section 5 is dedicated to the computational results, as well as the comparison of the performance and sensitivity analysis of the proposed algorithms. Finally, in section 6, conclusions and suggestions of future research are made.

2. Problem Definition

First of all, it is better to have a clear definition of the network elements used in this paper. Here, the term "link" is used for all streets and roads of all types in the network. When the link is two-way, it consists of two "arcs", and when it is one-way, it has one arc. All arcs are directed, meaning that all the vehicles, including buses and automobiles, on the same arc moves to the same direction. The capacity of each arc is defined by the number of lanes on that arc. For two-way links, lanes are found on both of the arcs. If the link is one-way, all lanes are allocated to the single existing arc.

In this paper, the design of an urban road network is proposed. The objective is to determine the optimal combination of one-way and two-way links, link and lane additions, and lane allocations on two-way links simultaneously, under a multi-objective decision making framework. Link expansions are actually considered as adding lanes to the existing network links. Indeed, this problem is a DNDP as all the decisions involved can be represented by discrete values.

The allocation of lanes to the two-way links can be performed in two ways, creating two variations of the problem: (1) Two-way links are not restricted to be symmetric and each direction can have different numbers of lanes; this problem variation is referred to as DNDP1. (2) Two-way links must be symmetric in terms of lane allocations in each direction; this problem variation is referred to as DNDP2. DNDP1 is a relaxed form of DNDP2 that only allows one-way or symmetric two-way links. DNDP1 is more likely to be observed in real urban networks, but DNDP2 can still be observed in reality too. Indeed, DNDP1 which allows the presence of non-symmetric two-way links in the network provides more flexibility in configuration and also improves performance measures.

Signal settings are not taken into account, similar to the problems studied by Yang and Bell (1998a; 1998b) and Yang and Wang (2002). They will be considered in future studies.

2.1. Assumptions

The main assumptions for the studied problems are listed below:

- The basic network exists in advance with all of the links as two-way.
- Travel demand between each origin and destination (OD) pair is fixed and known.
- Route selection behavior of network users is based on the user equilibrium principle.

2.2. Inputs

The following are the required inputs:

- The travel demand matrix, i.e. the total travel demand between each OD pair.
- The maximum available investment budget set by the network authority for the development projects.

- The parameters of the transportation network such as the free flow travel time and capacity of each arc, the number of lanes on each arc, etc., as well the arc performance functions and their associated parameters.
- Possible link construction projects defined by the network authority. This makes the candidate link construction project set.
- Maximum possible number of lanes to be added to each side of existing links, which is defined by the network authority. The available vacant land adjacent to each link influences on the maximum possible number of added lanes on them. Lane addition projects involve adding equal number of lanes on each side of the link.

2.3. Outputs

The two problems have the same following outputs:

- The new links that should be constructed.
- The number of lanes that should be added to each existing link.
- The allowable direction of traffic movement on each link (i.e. the set of two-way links and the set of one-way links).
- The number of lanes on each arc of streets.
- At last, the maximum level of travel demand that can be tackled by the network after executing the network improvement decisions.

Both DNDP1 and DNDP2 have three objective functions. The first objective function is defined by the reserve capacity of the network, which is measured by the greatest multiplier that can be applied to the current travel demand matrix, in a way that the increased traffic flow on any arc does not exceed the corresponding arc capacity. The multiplier is a common coefficient that is applied to the current travel demand of each OD pair in the network. The greatest multiplier μ^* is usually above 1.0 for a network with a normal or low level of congestion, and below 1.0 for a highly congested network with insufficient of network capacity. When $\mu^* > 1$, the network is said to have a reserve capacity amounting to $100(\mu^* - 1)\%$ of the existing OD matrix. However, when $\mu^* \leq 1$, the network has no reserve capacity. This objective function is concerned with the objective of the network authority, which is to maximize this multiplier and hence, reserve capacity.

As mentioned earlier, the maximization of reserve capacity can avoid the occurrence of capacity paradox at any demand level, while at the same time it considers the future conditions of the network with increased demand levels. The maximization of the reserve capacity can only minimize the total travel time to a certain extent when the congestion level is high (Yang and Wang, 2002).

Pursuing this fact, we propose a second travel time related objective function: the average ratio of the minimum OD travel time to the corresponding minimum free flow travel time. The aim is to minimize the average congestion level on the travel routes between OD pairs, rather than the total or average travel time for the fixed number of network users. The average congestion level takes the distance or minimum travel time between origin and destination into account while the average travel time does not. Using total or average travel time as the performance measure will thus give a bias to improve not very congested routes between the OD pairs with high minimum travel time, but will give a lower preference to improve very congested routes between the OD pairs with low minimum travel time. Indeed, minimizing the average ratio can avoid this inequity problem.

Another reason is to consider both the future and the present network conditions while designing the network. Maximizing the reserve capacity, which considers the future conditions of the network, may result in higher travel times and congestion levels at the current demand level.

The third objective function is about the difference between forward and backward minimum travel times of all OD pairs and the goal is to minimize the maximum difference. This avoids a severe imbalance of the minimum travel times in the two opposite directions between the same OD pair for the users frequently traveling back and forth between the origin and destination. Transportation system users usually expect that the two minimum travel times between an origin and destination to be about the same.

Obviously, the second and the third objective functions are related to the concern of the network users, whereas the first objective function is related to the concern of network authority. The authority is benefited from a less need of implementing network capacity expansion projects in the future, if they succeed in providing sufficient extra capacity to tackle future demand levels. The network authority will also be benefitted from receiving less opposition to the network design since the concern of network users is also considered.

3. Mathematical models and notations

The following notations are used in the model formulation.

3.1 Sets

N: set of network nodes.

A: set of existing network arcs.

A': set of candidate network arcs.

L: set of existing network links.

L': set of candidate network links.

S_l: set of arcs for each existing network link l.

$S_{l'}$: set of arcs for each candidate network link l'.

W: set of all OD pairs.

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

3.2 Decision Variables

μ: demand matrix multiplier.

y_l: number of lanes added to each side of 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 lanes allocated to arc (i, j).

x_{ij}^*: user equilibrium traffic flow on arc (i, j) for the increased demand level.

$\bar{x}_{ij}^*$: user equilibrium traffic flow on arc (i, j) for the current demand level.

X_r: user equilibrium flow on route $r \in R$ for the increased demand level.

$\bar{X}_r$: user equilibrium flow on route $r \in R$ for the current demand level.

3.3 Parameters

d_{pq}: travel demand between OD pair (p, q).

D: $[d_{pq}]$: matrix of travel demands.

B: total budget for lane addition and link construction projects.

$y_l^{\max}$: maximum number of lanes that can be added to each side of existing link l.

K_l: current number of lanes on existing link l.

$K_{l'}$: number of lanes on new link l' if it is built.

M: a large positive number.

3.4 Functions

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

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

t_{ij}: travel time function of arc (i, j).

$c_{ij}(k_{ij})$: capacity of arc (i, j) (which equals the product of the number of lanes k_{ij} on the arc and the capacity of a lane).

$T_{p'q'}$: minimum travel time between nodes p' and q'.

Z_1: the first objective function - demand matrix multiplier.

Z_2: the second objective function - the average ratio of the OD travel time to the corresponding free flow travel time.

Z_3: the third objective function - maximum difference between forward and backward minimum travel times.

$\bar{v}_{pq}$: minimum travel time between OD pair (p,q) for the current demand level.

$\bar{v}_{pq}^0$: minimum free flow travel time between OD pair (p,q).

δ_{ijr} : binary variable for the increased demand level, which equals 1 if route *r* uses arc (*i*, *j*), and zero otherwise.

$\bar{\delta}_{ijr}$: binary variable for the current demand level, which equals 1 if route *r* uses arc (*i*, *j*), and zero otherwise.

Figure 1 indicates example of two links and their modeling with decision variables. At the right hand side, a one-way existing lane link with two lanes is indicated. At the left hand side, there is a two-way link with two existing (in white) and two added lanes (in gray), and the lane allocation is asymmetric (i.e. one and three lanes on each side). The corresponding values of the variables are indicated for each link.

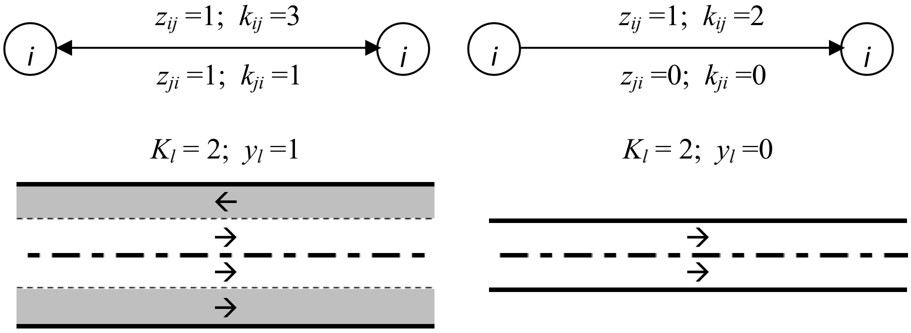

Figure 1. Examples of link modeling

3.5 Model for DNDP1

DNDP1 is a problem with two levels: the one upper level problem represents the decision making problem of the network authority, and two lower level problems represent the network users' behavior in choosing travel routes, one for the current demand level and another for the scaled demand level. The upper level problem is a tri-objective mixed integer mathematical model. The lower level problems compute the corresponding equilibrium flows and travel times for each set of decision variable values in the upper level problem (i.e. a network design scenario). The lower level problems are by their nature non-linear minimization problems represented as variational inequalities in the upper level problem. Mathematically, DNDP1 is formulated as follows:

$$\text{Max } Z_1 = \mu \tag{1}$$

$$\text{Min } Z_2 = \frac{\sum_{(p,q)\in W} \frac{\bar{v}_{pq}}{\bar{v}_{pq}^0}}{|W|} \tag{2}$$

$$\text{Min } Z_3 = \underset{(p,q)\in W}{\text{Max}} (\bar{v}_{pq} - \bar{v}_{qp}) \tag{3}$$

Subject to

$$\sum_{l\in L} g_l(y_l) + \sum_{l'\in L'} g'_{l'}(y_{l'}) \leq B \tag{4}$$

$$0 \leq y_l \leq y_l^{\max} \qquad \forall l \in L \tag{5}$$

$$k_{ij} + k_{ji} = K_l + 2y_l \quad \forall l \in L, (i,j) \in S_l \tag{6}$$

$$k_{ij} + k_{ji} = K_{l'}.u_{l'} \qquad \forall l' \in L', (i,j) \in S_{l'} \tag{7}$$

$$x_{ij}^{*} \leq c_{ij}(k_{ij}) \qquad \forall (i,j) \in A \cup A' \tag{8}$$

$$z_{ij} \leq k_{ij} \qquad \forall (i,j) \in A \cup A' \tag{9}$$

$$k_{ij} \leq M.z_{ij} \qquad \forall (i,j) \in A \cup A' \tag{10}$$

$$z_{ij} + z_{ji} \geq 1 \qquad \forall (i,j) \in A \tag{11}$$

$$z_{ij} + z_{ji} \geq u_{l'} \qquad \forall (i,j) \in A', l' \in L' \tag{12}$$

$$z_{ij} + z_{ji} \leq M.u_{l'} \qquad \forall (i,j) \in A', l' \in L' \tag{13}$$

$$T_{p'q'} \leq M \qquad \forall p', q' \in N \tag{14}$$

$$\mu \geq 0 \tag{15}$$

$$k_{ij}, y_l \geq 0, \text{ are integers} \quad \forall (i,j) \in A \cup A', l \in L \tag{16}$$

$$z_{ij}, u_{l'} \in \{0,1\} \qquad \forall (i,j) \in A \cup A', l' \in L' \tag{17}$$

$$\sum_{(i,j)\in A\cup A'} t_{ij}\left(x_{ij}^{*}, c_{ij}\right).\left(x_{ij} - x_{ij}^{*}\right) \geq 0 \quad \forall x_{ij} \in \Omega \tag{18}$$

$$\Omega = \left\{ x_{ij} \middle| \begin{array}{c} \sum_{r\in R_{pq}} X_r = \mu.d_{pq} \;\; \forall (p,q) \in W; \\ x_{ij} = \sum_{(p,q)\in W} \sum_{r\in R_{pq}} X_r.\delta_{ijr} \;\; \forall (i,j) \in A \cup A'; \\ X_r \geq 0 \end{array} \right\} \tag{19}$$

$$\sum_{(i,j)\in A\cup A'} t_{ij}\left(\bar{x}_{ij}^{*}, c_{ij}\right).\left(\bar{x}_{ij} - \bar{x}_{ij}^{*}\right) \geq 0 \quad \forall \bar{x}_{ij} \in \bar{\Omega} \tag{20}$$

$$\bar{\Omega} = \left\{ \bar{x}_{ij} \middle| \begin{array}{c} \sum_{r\in R_{pq}} \bar{X}_r = d_{pq} \;\; \forall (p,q) \in W; \\ \bar{x}_{ij} = \sum_{(p,q)\in W} \sum_{r\in R_{pq}} \bar{X}_r.\bar{\delta}_{ijr} \;\; \forall (i,j) \in A \cup A'; \\ \bar{X}_r \geq 0 \end{array} \right\} \tag{21}$$

The first objective maximizes the reserve capacity. The second one minimizes the average ratio of the minimum OD travel time to the corresponding minimum free flow travel time, and the last one minimizes the maximum difference between forward and backward minimum travel times. Constraint (4) imposes a budget limit to the total cost of lane additions and link constructions. Constraint (5) limits the number of lanes added to the existing network links. Constraints (6) and (7) determine the allocation of lanes on the existing links and the new links respectively. Constraint (8) is the principle constraint of reserve capacity maximization where the traffic flow on each arc cannot exceed the corresponding arc capacity. Constraints (9) and (10) assure that no capacity is assigned to non-existing arcs. Constraint (11) guarantees that at least one arc is present for each existing link. In other words, blocking of streets is not allowed. Constraints (12) and (13) ensure that at least one arc is built for each new link. Constraint (14) enforces that the minimum travel time between each pair of nodes to be less than a very large value. This can ensure that there is at least one path between each OD pair of the network and that the network is strongly connected. Because when there is no path connecting an OD pair, the OD travel time must be very large. Constraints (15)-(17) define the domains of variables. Constraints (18)-(19) and constraints (20)-(21) present the variational inequality formulations of the user equilibrium traffic assignment as the two lower level problems. They determine the user equilibrium traffic flows for the increased and current travel demand level respectively.

3.6 Model for DNDP 2

The mathematical model for DNDP2 is obtained by adding the following constraints to the previous model:

$$k_{ij} \leq k_{ji} + M.\left(1 - z_{ji}\right) \quad \forall (i,j) \in A \cup A', \text{ and} \tag{22}$$

$$k_{ji} \leq k_{ij} + M.\left(1 - z_{ij}\right) \quad \forall (i,j) \in A \cup A' \tag{23}$$

The above constraints assure that whenever a link is two-way, the number of assigned lanes on each direction must be equal.

The minimax objective function in the above models can be linearized by a simple transformation, and adding two constraints to the upper level problem: Objective (3) can be written as objective (24) and the constraints (25)-(26).

$$\text{Min } F \tag{24}$$

$$\underset{(p,q)\in W}{\text{Max}} (\bar{v}_{pq} - \bar{v}_{qp}) \leq F \tag{25}$$

$$F \geq 0 \tag{26}$$

4. Solution Procedures

It is important to note that even a simple bi-level program has been proved to be NP-hard (Ben-Ayed *et al.* 1988). Also, even if the both upper and lower levels of the problem are convex, the whole problem may not be convex. There are some exact methods to solve the problem for obtaining global optimum solutions. But the fact is that, they cannot be applicable for large and real size transportation networks. Meta-heuristics have been largely applied to obtain good solutions for realistic, large network design problems, especially in recent years. Hence, one of the current directions is to adopt meta-heuristic techniques for bi-level programs. Since the proposed problems are new, and there are many constraints needed to be handled, directly applied existing metaheuristic is not possible. Hence, we propose some well-organized solution algorithms related to meta-heuristics in this paper. These methods can result in good solutions even for large networks as will be presented in the following sections.

For the two variations of the problems, three hybrid multi-objective population-based metaheuristic algorithms are proposed, namely:

- A multi-objective Hybrid Genetic Algorithm (mHGA),
- A multi-objective Evolutionary Simulated Annealing (mESA),
- A new multi-objective Hybrid Artificial Bee Colony algorithm (mHABC).

After exploring the literature, we conclude that mESA and mHABC have never been used to solve urban transportation network design problems. The multi-objectivity of the two problems is captured by the Adaptive Weight Approach (AWA) that utilizes useful information from the population solutions to assign weights to each objective function and to obtain a single ranking for each solution (Gen and Cheng, 2000).

In the following, the similarities and the details of the three algorithms are presented. In particular, in section 4.1 a comparison is organized between them and the details of methods are described in sections 4.2-4.4.

4.1. Similarities and Comparisons of the Proposed Algorithms

The three hybrid meta-heuristics have some concepts in common. These similarities are the solution encoding, fitness function calculation, checking out the feasibility of solutions, generating the initial population, and building the Pareto-optimal solutions set.

Solution Encoding

As can be noticed from the mathematical formulation, the upper level decision variables, including link construction, and lane addition and allocation, are discrete. The continuous variables are just related to the two lower level problems, which can be obtained by solving those problems, having the fixed values of discrete variables.

Each chromosome represents the complete configuration of the candidate and existing links in the network. Chromosomes are represented by 2-row matrices, in which columns correspond to network links and rows correspond to their lane allocations. The total length of each chromosome is $L+L'$, which is equal to the total number of existing and candidate links respectively. For each column (i.e. link), the lane allocation is indicated by the non-negative integers in the last two row. An example network with 7 nodes and 12 links and its corresponding chromosome are depicted in figures 2 and 3, respectively. There are two possible links to be constructed, i.e. 6-2 and 4-7. Solid lines indicate that links are present in the design scenario and the dotted and dashed line indicates that link 4-7 is not chosen to be constructed. The arrow heads point in the allowed direction of movement in each link.

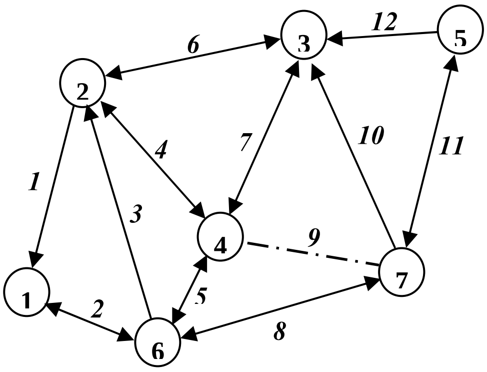

Figure 2. A typical network

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

Figure 3. The chromosome representation

Calculation of Fitness Values

As pointed out earlier, the fitness values are defined by rankings obtained by the adaptive weight approach. The adaptive weight for each objective function is calculated by the following equation:

$$w_o = \frac{1}{Z_o^{\max} - Z_o^{\min}} \quad \text{where} \quad o = 1,2,3 \tag{27}$$

where $Z_o^{\max}$ and $Z_o^{\min}$ are respectively the maximum and the minimum values of objective function o among the population solutions. Since the original form of the fitness function in the adaptive weight approach is proposed for all maximization objective functions, in this study the minimization objective functions of (2) and (3) are converted to maximization objective functions by multiplied negative one to their original objective functions. The fitness value of a solution is calculated as follows:

$$Z(s) = \sum_{o=1}^{3} w_o . (Z_o(s) - Z_o^{\min}) \tag{28}$$

The adaptive weights are updated each time when the population is modified, and the fitness function values are recalculated for the population solutions. Higher fitness values mean better solutions. The values of the objective functions are computed by the solution procedures described below.

The calculation of the reserve capacity in the first objective function is undertaken by solving a sub-problem, in which the values of discrete variables have come to be fixed previously. The corresponding sub-problem is defined as follows:

Given y_l^*, $u_{l'}^*$, z_{ij}^* and k_{ij}^*, determine μ and x_{ij}^* to:

$$\text{Max } \mu \tag{29}$$

Subject to

$$x_{ij}^* \leq c_{ij}(k_{ij}^*) \quad \forall (i,j) \in A \cup A' \tag{30}$$

$$\mu \geq 0 \tag{31}$$

$$\sum_{(i,j)\in A\cup A'} t_{ij}\left(x^*_{ij}, c_{ij}\right).\left(x_{ij} - x^*_{ij}\right) \geq 0 \quad \forall x_{ij} \in \Omega \tag{32}$$

$$\Omega = \left\{ x_{ij} \middle| \begin{array}{c} \sum_{r\in R_{pq}} X_r = \mu . d_{pq} \;\; \forall (p,q) \in W; \\ x_{ij} = \sum_{(p,q)\in W} \sum_{r\in R_{pq}} X_r . \delta_{ijr} \;\; \forall (i,j) \in A \cup A'; \\ X_r \geq 0 \end{array} \right\} \tag{33}$$

At the first sight, it seems that the above sub-problem can be solved by sensitivity analysis based (SAB) algorithm which is commonly employed in solving network design problems with continuous variables (Tobin and Friesz, 1988). The drawback of SAB algorithm is that this method is not applicable for solving all types of problems because of the nonexistence of inversions of the matrices in some cases. Even the newer versions of the SAB algorithm has been developed, they are not successful in solving some types of bi-level problems with user equilibrium constraints. Therefore, we develop our heuristic to solve the above sub-problem by using the special structure of the above sub-problem.

The idea of our heuristic is to find a rough interval for the optimal value of μ and, then to use a one-dimensional line search method to find the optimal value in that interval. For each value of μ, the user equilibrium assignment problem (32)-(33) is solved to determine the optimal x_{ij} values. The variational inequality (32)-(33) is in fact a convex nonlinear problem, which can be exactly solved by many methods including a convex-combination-based algorithm called the Frank-Wolfe (FW) method (see Sheffi (1985) for the details) that is often used in solving user equilibrium problems.

Let *LB* and *UB* be the lower and upper bounds of the rough interval, respectively. The heuristic is then described below:

```
Initialization 1: Set $\mu = 0$, $LB = 0$, $UB = \infty$
Phase 1: Repeat until the value of UB is changed
    - $\mu = LB + 0.5$
    - Solve the variational inequality (32)-(33) by the FW method to obtain user equilibrium values
      for $x_{ij}$
    - If constraint (30) is satisfied, set $LB = \mu$; otherwise set $UB = \mu$
Initialization 2: Set $\mu_a = LB$, $\mu_b = UB$
Phase 2: While $\mu_b - \mu_a > \varepsilon$, where $\varepsilon$ is the tolerance for the convergence,
    - Set $\mu = (\mu_a + \mu_b) / 2$
    - Solve the variational inequality (32)-(33) by the FW method to obtain user equilibrium values
      for $x_{ij}$
    - If constraint (30) is satisfied, set $\mu_a = \mu$; otherwise set $\mu_b = \mu$
Output: Return the value of $\mu = (\mu_a + \mu_b) / 2$
```

In the above, the increment value in phase 1 and also the length of the rough interval for μ are set to 0.5. This value is selected based on 1) some preliminary experiments showing that 0.5 is the most proper increment value, and 2) the fact that the reserve capacity of a real network is usually less than 10. Moreover, ε is set to 0.01, which is believed to be accurate enough given that the reserve capacity measure is seldom greater than 10.

The second and the third objective values are calculated based on the minimum travel times obtained from solving the variational inequality (20)-(21) for the current demand level. The calculation procedure is as follows:

- Solve the variational inequality (20)-(21) by the FW method to obtain user equilibrium values for $\bar{\mathbf{x}}_{\mathbf{ij}}$.
- Find the shortest paths for the actual travel times and for the free flow travel times, and thus the minimum travel time $\bar{v}_{pq}$ and the minimum free flow travel time $\bar{v}_{pq}^{0}$ for each OD pair.
- Calculate the second objective value by finding the average of $\bar{v}_{pq} / \bar{v}_{pq}^{0}$ of all OD pairs.
- Calculate the third objective value by finding the maximum of $(\bar{v}_{pq} - \bar{v}_{p}^{0})$ of all OD pairs.

Checking for Feasibility of Solutions

Two types of feasibilities are checked for each solution; (a) budget feasibility of construction and lane addition projects, (b) strong connectivity of the network (There is at least one directed path connecting each node in the network to every other nodes). The budget infeasibility is checked using constraint (4) which is straightforward. The strong connectivity is checked using a two-stage procedure.

- Stage 1: Checking whether all nodes have at least one incoming and one outgoing lane: If all the nodes have at least one incoming and one outgoing lane, then proceed to stage 2; otherwise discard the network and stop. This stage is based on the necessary condition for strong connectivity of the network. If there is no incoming (no outgoing) lane for a node, then no flow can reach (depart from) that node, and the network must violate the strong connectivity requirement.
- Stage 2: Checking whether a shortest path exists between all OD pairs: determine whether there is a shortest path for each OD pair by using the algorithm of Dijkstra. If there is a shortest path for every OD pair, the network satisfies the strong connectivity condition. Otherwise, the network violates the condition. The reason is as follows. If there is a shortest path between an OD pair, there must be at least one directed path connecting this OD pair. If not, the origin is disconnected from the destination, and this OD pair violates the strong connectivity condition.

This two-stage mechanism helps the algorithm to discover the disconnected networks at the early stage and reduce the frequency of using Dijkstra's algorithm for connectivity checking.

The budget infeasibility is possible to be repaired using the procedures explained in the next sections, but repairing the connectivity infeasibility is an unpractical and complex task. Thus, each disconnected network is discarded instantly.

Initial Population Generation

The construction of the initial population is performed using a 3-phase heuristic procedure. To generate one initial solution, the procedure depicted in figure 4 is used.

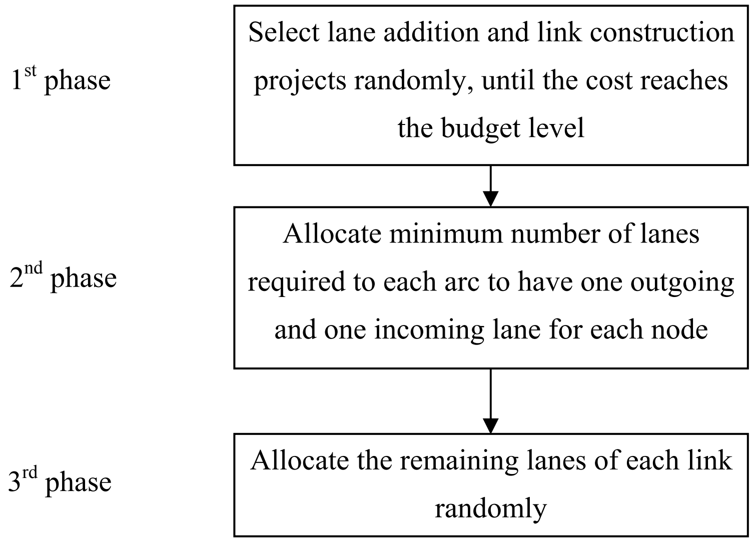
1st phase
Select lane addition and link construction projects randomly, until the cost reaches the budget level
2nd phase
Allocate minimum number of lanes required to each arc to have one outgoing and one incoming lane for each node
3rd phase
Allocate the remaining lanes of each link randomly

Figure 4. Initial solution generation procedure

First, the possible expansion or construction projects are selected randomly, until the construction cost reaches the predefined budget. Next, set of lanes are allocated randomly to each arc. In order to reduce the possibility of generating an infeasible, disconnected network, at the beginning, lanes are allocated in such a way that each node has at least one outgoing and one incoming lane. This can guarantee the network generated satisfying the necessary condition for strong connectivity. Then, the remaining lanes are allocated randomly to the network links. This technique saves computational time because the procedure reduces the chance of generating disconnected networks.

Then, stage 2 of strong connectivity procedure is carried out. If the generated network satisfies the strong connectivity requirement, then the initial solution is feasible. Otherwise, the solution is discarded. The 3-phase procedure is repeated until enough number of feasible solutions is generated.

Pareto Optimal Solution Set Generation

Like many multi-objective metaheuristics, a set of Pareto-optimal solutions is defined to maintain the list of non-dominated solutions in algorithms. The initial Pareto-optimal set is built after generating the initial population solution. All initial population members, which are not dominated by any of the other population solutions, are added to the initial Pareto-optimal set.

Whenever a new and feasible solution is generated by the algorithm, the solution is checked for Pareto-optimality. To do this, the following steps are carried out:

- If the solution is not dominated by any of the Pareto-optimal set members, then it is inserted in the list of non-dominated solutions.
- Any member of Pareto-optimal set that is dominated by the newly added member is removed from the list.

Comparison of the Proposed Algorithms

Table 2 compares the similarities and differences of the three proposed algorithms, and provides an overall view of these algorithms.

Table 2. Comparison of the developed algorithms

Algorithm	# of iterations	Solution generation method	Evolution strategy
mHGA	*G* generations	Selecting two parents, applying crossover, and applying SA on some offspring solutions	Replacing a number of population solutions with some offspring solutions, using the evolution mechanism
mESA	*G* generations	Selecting one parent and applying SA	Replacing some population solutions with the generated solutions set, using the evolution mechanism
mHABC	*C* cycles with two inner loops	**Inner loop 1:** Iteratively selecting each population solution once, and applying SA **Inner loop 2:** Iteratively selecting a population solution using the roulette wheel selection rule and applying SA	Replacing each selected population solution with its corresponding generated solution (in both loops)

4.2. Multi-objective Hybrid Genetic Algorithm

The first hybrid meta-heuristic we developed in this study is the multi-objective genetic algorithm. This is a population-based meta-heuristic based on GA proposed by Holland (1975). GA is grounded on the ideas of genetic and natural selection. GA and its extensions have been widely used as a solution method for solving the NDPs and have been pointed out as a successful method to obtain good solutions (e.g., Drezner and Salhi, 2002; Drezner and Wesolowsky, 2003; Cantarella *et al.*, 2006; Cantarella and Vitetta, 2006; Chen *et al.*, 2006; Poorzahedy and Rouhani, 2007).

The algorithm steps are 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:
  - o Select the first parent from the population by applying the roulette wheel rule.
  - o Select the second parent randomly from the Pareto-optimal set.
- Merge the selected parents to produce an offspring set using a crossover operator.
- 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 repair the solution using the budget reduction method if necessary.
- Calculate the fitness value for each offspring solution.
- Update the Pareto-optimal set using the offspring set.
- Update the current population using the evolution mechanism:
  - o Eliminate the population solutions and insert the offspring solutions into the population.
  - o Apply SA on a portion of the best selected offspring solutions and update the population.
- Recalculate the adaptive weights and fitness values of population solutions.

Crossover Operator

The operator used here for crossover is adopted from Drezner and Wesolowsky (1997, 2003) with a modification. They used a process for merging parents in a way that their network structure was taken into account. For exploiting the structure, a connected set of links is chosen to form the offspring. This is performed by firstly defining a node named pivot node. Then, a count is assigned to each link of the network, including the set of new links. Links that are directly connected to the pivot node get count 1 and links that are connected to links with count 1 get count 2 and so on. If a link is connected to links with different counts, the minimum count value will be used to define the count of the link. Figure 5 depicts an example of assigning counts to links. In this figure, node 1 is selected as the pivot node.

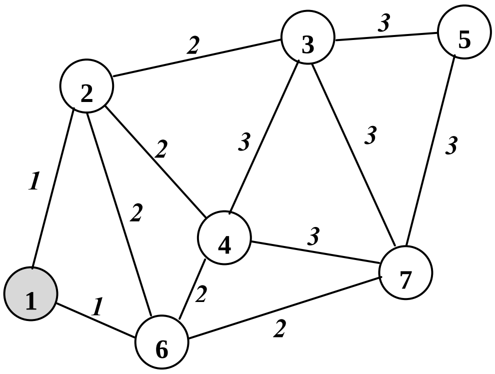

Figure 5. A typical link count assignment

After assigning link counts, the median of all counts is computed. All new and existing links of the first offspring with counts below the median get their design (i.e. lane numbers, and lane allocation,) from the first parent and the new and existing links with their counts above the median get their design from the second parent. Other links with counts equal to median can be chosen randomly from one of the parents.

The modification imposed to the original method is that the whole set of links with the count equal to median must be chosen from just one parent. In the original version of the operator, the design of each of such links can be taken from either parents randomly and independent from other links. The modified method will reduce the possibility of creating disconnected networks, since this method ensures a group of links to be chosen from a single parent.

Since each node can be selected as a pivot node and each offspring is uniquely defined for a given pivot node, the total number of offspring generated equals the number of network nodes $|N|$.

Each offspring generated is duplicated and then for each copied offspring the one-way links with count higher than median are reversed. This can be easily done because of having the link counts of solution generated from each pivot node. In this way, another $|N|$ offspring are produced, leading to a total of $2|N|$ offspring produced from this procedure.

Offspring Feasibility

Each produced offspring is examined for feasibility; if it is infeasible in terms of network connectivity, it is discarded. After that, the offspring is checked for total construction cost. If the offspring violates the budget constraint, a budget reduction sub-routine is applied to repair the solution. The sub-routine is as follows:

- Check every possible swap of lane addition or link construction projects.
- If there is at least one feasible swap in terms of both budget and the network connectivity, select a random swap and apply it.
- If there is no feasible swap, then check whether some projects can be excluded from the solution.
- If there is at least one project that can be excluded without violating the strong connectivity requirement, then repeat the following steps until the budget level is not violated or the network becomes disconnected:
    - Exclude the project with the maximum cost.
    - If the network becomes disconnected, then stop and report infeasibility.

Embedded SA Algorithm

A short procedure of Simulated Annealing (SA) algorithm is applied on a number of offspring solutions. This approach offers two advantages; first, it provides a series of random perturbations to the offspring which can be a substitute of mutation operator. Second, it acts as a heuristic training method that provides better solutions. The SA algorithm is applied to a portion of offspring solutions rather than all of them. This is because of the high computational requirements for large or even medium sized networks. The SA algorithm used here performs only one iteration for each temperature. Each move in SA is performed as the following:

- For each link, randomly decide to change it, or keep it as it is.
- For each link to be changed, select a random lane allocation.

In changing the lane allocation of a link, a feasibility interval is used. This interval indicates the possible lane allocation schemes which applying them do not lead to zero incoming or outgoing lanes for any network node. By using this interval, one can assure that at least the necessary condition for the strong connectivity of the network is satisfied. This can help decrease the probability of the occurrence of disconnectedness of the network.

For the link l between nodes i and j, the feasibility interval is defined as the minimum and maximum number of lanes that can be chosen for arc (i, j) such that after applying the change in lane allocation, the number of outgoing and incoming lanes for nodes i and j are all greater than zero. The interval is defined as $[LB_{ij}',UB_{ij}']$. It must be noted that considering lanes on arc (i, j) suffices, since the number of lanes on arc (j, i) can be easily deduced from K_l.

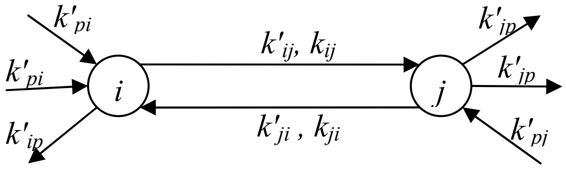

Figure 6. Variables and parameters for feasibility interval

Figure 6 illustrates the variables and parameter used to define the feasibility interval. As depicted in figure 6, both arcs (i, j) and (j, i) need two notations; $k'ij$ and $k'ji$ for the current lane allocation (as parameters) and k_{ij} and k_{ji} for the new lane allocation (as variables). In order to consider the number of outgoing and incoming lanes for nodes i and j, it is required to compute the total number of incoming and outgoing lanes for all related arcs to i and j, except arcs (i, j) and (j, i). Thus, k'_{ip}, k'_{pi}, k'_{pj}, k'_{jp} are defined as the current lane numbers of the related outgoing or incoming arcs, which are treated as fixed (parameters). Now to calculate the interval, one needs to consider the following constraints:

$$k_{ij} + \sum_{p\in\Phi_i,p\neq j} k'_{ip} \geq 1 \tag{34}$$

$$k_{ji} + \sum_{p \in \Phi_j, p \neq i} k'_{jp} \geq 1 \tag{35}$$

$$k_{ji} + \sum_{p \in \Phi_i, p \neq j} k'_{pi} \geq 1 \tag{36}$$

$$k_{ij} + \sum_{p \in \Phi_j, p \neq i} k'_{pj} \geq 1 \tag{37}$$

$$k_{ij} + k_{ji} = k'_{ij} + k'_{ji} \tag{38}$$

where Φ_i and Φ_j are the sets of nodes that are adjacent to nodes i and j. Constraints (34)-(35) imply that the total number of outgoing lanes from nodes i and j must be greater than one. Constraints (36)-(37) imply that the total number of incoming lanes to nodes i and j must be greater than one. Finally, constraint (38) is the lane allocation equation which requires that the total number of lanes between nodes i and j should not change. The summation terms in constraints (34)-(37) indicate the total number of incoming or outgoing lanes to each node, which are fixed (or treated as parameters). k_{ij} and k_{ji} are the only variables of the constraint set.

From the above inequalities, one can extract the maximum and minimum possible values for k_{ij}, i.e. LB_{ij}' and UB_{ij}'. LB_{ij}' can be obtained using constraints (34)-(35), and UB_{ij}' can be obtained using constraints (37)-(39). Relations (39)-(40) indicate the formulas of the two bounds.

$$LB'_{ij} = \max\left\{0, \max\left\{1 - \sum_{p \in \Phi_j, p \neq i} k'_{pj}, 1 - \sum_{p \in \Phi_i, p \neq j} k'_{ip}\right\}\right\} \tag{39}$$

$$UB_{ij}^{'} = \min\left\{k'_{ij} + k'_{ji}, \min\left\{\sum_{p \in \Phi_j} k'_{jp} + k'_{ij} - 1, \sum_{p \in \Phi_i} k'_{pi} + k'_{ij} - 1\right\}\right\} \tag{40}$$

When k_{ij} is selected randomly from the interval determined by (39) and (40), k_{ji} can be obtained from the total number of lanes of the link. Note that the above interval is valid for DNDP1. For DNDP2, symmetry restrictions must be taken into account. This can be done by choosing the feasible number of lanes for one-way or symmetric two-way lane allocations from the values provided by the interval.

After changing lane allocations for all selected links, the solution is checked for connectivity feasibility; all random lane allocation changes are discarded if the network is disconnected, and the procedure is applied again based on the current lane allocation. If no feasible solution is found after repeating the procedure up to 10 times, then the current temperature (i.e., iteration) is skipped. This is set so as to keep the computational effort rational.

Evolution Mechanism

The evolution process is undertaken by replacing some of the population solutions with the generated offspring solutions with higher fitness values. Let q be the parameter of the algorithm and be a real number selected between 0 and 1 so that a trade-off is made between computational effort and solution quality. (A small value leads to a lower computation effort but a lower solution quality, and vice versa.) Then, the process can be described as follows:

- Find the set of population solutions S' that are worse than at least one offspring solution.
- If $|S'| \geq |S_0|$, where S_0 is the set of offspring solutions, then select $q|S_0|$ best offspring solutions to undergo the short SA process. Each of the selected offspring solutions will give one solution. If a new solution obtained is better than the selected one, then update the corresponding solution in S_0. Otherwise, keep the selected offspring as it is. Then, use the modified S_0 solutions, to replace $|S_0|$ worst population solutions.
- If $|S'| < |S_0|$, then select $q|S'|$ best offspring solutions to undergo the short SA process. Each of the selected offspring solutions will give one solution. If the new solution obtained is better than the selected one, then update the corresponding offspring solution. Otherwise, keep the offspring as it is. Then, use the best $|S'|$ offspring solutions to replace the worst $|S'|$ population solutions.

4.3. Multi-objective Evolutionary Simulated Annealing

Like genetic algorithm, simulated annealing is also one of the most prevalently used methods of solving network design problems (Ben-Ayed *et al.*, 1988; Friesz *et al.*, 1993; Lee and Yang, 1994; Drezner and Wesolowsky, 1997; 2003; Yang and Wang, 2002; Poorzahedy and Abulghasemi, 2005; Poorzahedy and Rouhani, 2007). Multi-objective Evolutionary Simulated Annealing (mESA) proposed in this paper is a hybrid version of SA and GA in a multi-objective framework. Population-based ESA is similar to GA but uses SA as the crossover and mutation operators. In every GA iteration, a sole parent is selected and a short version of SA is applied to generate offspring. The two advantages of ESA have been discussed in Aydin and Fogarty (2004). The multi-start property of ESA provides a more uniform distribution of random moves along the whole procedure and helps to diversify the solutions. The other advantage is its evolutionary approach. Before the details of ESA are discussed, the entire procedure of the algorithm is described below.

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

Phase 2: Repeat the following procedure for G generations:

- Randomly select a parent solution either from the population using the roulette wheel rule or from the Pareto-optimal set.
- Apply SA on the selected parent to obtain an offspring set:
  - Initialize *Temperature*.

    - While *Temperature* is greater than a specified value, repeat:
        - Generate a random solution by applying perturbations in lane allocations, and by swapping a present lane addition or link construction project, with another one.
        - Check the connectivity of the solution and discard the solution if the network becomes disconnected.
        - Calculate the fitness value of the solution.
        - Add the solution to the offspring set and update the Pareto-optimal set.
        - If the random solution is better than the current solution, substitute the current solution by the random solution; Otherwise, accept the random solution by a probability of $e^{-\Delta/Temperature}$ (Δ is the difference of the objective values).
        - Reduce *Temperature*.
    - Return all the generated solutions.
- Replace the population solutions with the generated solutions using the evolution mechanism.
- Recalculate the adaptive weights and fitness values for population solutions.

mESA has the solution encoding, fitness value calculation, initial population generation, and the evolution mechanism procedures in common with mHGA. Like Aydin and Fogarty (2004), a short version of SA is substituted of the crossover and mutation in the proposed mESA. The embedded SA is similar to the SA in mHGA, except that an additional project swap procedure is applied when producing a random solution. In this procedure, at first every possible swap of lane addition or link construction projects is considered. A possible swap means that the budget constraint is not violated, and also the network doesn't become disconnected by applying the swap. After examining all possible swaps, one is selected randomly to be applied.

If no possible swap is found, another approach is adopted; all selected projects are checked for cancellation and among the possible cancellation, one project is selected for cancellation.

4.4. Multi-objective Hybrid Artificial Bee Colony Algorithm

Artificial Bee Colony algorithm (ABC), first introduced by Karaboga (2005) is a swarm-based metaheuristic. It is inspired by the intelligent behaviors, including collective, interactive and self-organized behaviors) of foraging bees for finding nectars. ABC and other bee algorithms have been developed to solve combinatorial optimization problems. These algorithms have some common features. First of all, bees explore and exploit the environment near their hive which is our search space. Second, they evaluate the nectar amounts at each location or solution visited, inform and attract other bees to select the qualified solutions, and search in their neighborhood. Third, these algorithms are conducted so that better sites are explored more.

One of the studies related to the bee algorithms was Teodorović and Dell'orco (2005) in which they examined the problem of transportation on ride-matching or ride-sharing. Other papers that proposed Artificial Bee Colony algorithms for solving their optimization problems are Yang (2005), Basturk and Karaboga (2006) and Karaboga and Basturk (2008). Chong *et al.* (2007) also proposed a Bee Colony solution method for job shop scheduling problems. The transportation problems which were solved by a bee colony algorithm so far were in the studies of Teodorović and Dell'orco (2005), Szeto *et al.* (2011), and Szeto and Jiang (2012). Nevertheless, these papers focus on one objective. Using ABC for solving multi-objective problems has been studied in other disciplines (see for example Pham and Ghanbarzadeh, 2007 and Low et al. 2011).

The basic characteristics of the proposed hybrid ABC is similar to the method, firstly introduced by Basturk and Karaboga (2006). Instead of working with partially built solutions, complete solutions are adopted. The colony consist of three groups of bees: employed bees, onlookers and scouts. Employed bees exploit available food sources that correspond to solutions in the optimization problem. These bees also gather required information and share them with the onlookers. Each onlooker bee chooses from a food source near to one of the food sources searched by an employed bee and evaluate the amount of nectar (corresponding to determining the objective value of a solution). The employed bee becomes a scout when the food source is abandoned, and the scout starts to search a new food source. This occurs when the solution has not been improved after a predetermined duration called limit.

In the ABC proposed in this paper, a small SA procedure is embedded like the works of Sadiq and Hamad (2010) and Alzaqebah and Abdullah, (2011). This SA procedure is used instead of the greedy solution procedure in Basturk and Karaboga (2006), because the former is found to be more effective. Also in the original ABC, a restriction of no more than 1 for scout at any time is imposed. This restriction is relaxed in our procedure. The SA used in this algorithm is similar the one developed for mESA.

In the following, the phases of the proposed algorithm are presented:

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

Phase 2: Repeat the following procedure for *C* cycles:

- Produce new solutions for employed bees by applying SA on all members of the population.
- For each onlooker, randomly select a solution based on the roulette wheel selection rule and apply SA on this solution.
- Find the abandoned solutions and replace them with random solutions obtained from Pareto-optimal set or with new solutions when the Pareto-optimal set is empty.
- Recalculate the adaptive weights and fitness values of population solutions.

5. Computational Results

5.1. Test problems and data

The investigated problem in this study is new, causing the absence of a benchmark to test our proposed algorithms. Therefore, we modified a number of common test cases found in NDP or traffic assignment papers, and made to allow us to test the effectiveness of the proposed solution methods. The test networks are given in table 3, which are classified into small, medium and large networks according to their size.

Table 3. Testing networks

Network size	Network adopted	Notation	Number of nodes	Number of links	Number of OD pairs
Small	The Harker and Friesz (1984) network	HF	6	8	2
	The Nguyen and Dupuis (1984) network	ND	13	19	4
	A reduced Sioux Falls network used in LeBlanc et al. (1975)	SF1	14	19	176
Medium	The Nagurney (1984) network	NA1	20	28	8
	The Nagurney (1984) network	NA2	22	36	12
	The basic Sioux Falls network used in LeBlanc et al. (1975)	SF2	24	37	528
Large	The Nagurney (1984) network	NA3	40	66	6

The network attributes needed for solving the problem are the link characteristics such as free flow travel time, capacity, and the number of lanes on each link for the existing or new links. In each of the original test networks, some of link characteristics are available but some are not. Therefore, we had to assign reasonable values to the missing parameters. The test case also needs the demand matrix, project settings, project cost, and budget available. Table 4 gives a summary for these test cases. In all test cases, it is assumed that expansion projects consist of adding one lane on each side of the link, and new constructible links have 4 lanes in total.

Table 4. Parameters selected for test cases

Network size	Notation	Demand values	Free flow travel times	Number of link lanes	Capacity of links	Project setting, cost & budget level
Small	HF	Original values	4-6 minutes [a]	2-4 to provide more flexibility [b]	7-15 [c]	Newly defined projects, set costs and budget using SF2 values, Budget = $ 2,300,000
	ND	Original values	Original values	Original values	Original values	Newly defined projects, set costs and budget using SF2 values, Budget = $ 3,500,000
	SF1	Original values from SF2	Original values from SF2	Original values from SF2	Original values from SF2	
Medium	NA1	Original values	4-8 minutes	2-4	300-1000 [c]	

	NA2	Original values			1000-2500[c]	
	SF2	Original values	Original values	Original values	Original values	Existing projects plus selected new links, used existing and new costs, Budget = $4,000,000
Large	NA3	Original values	4-8 minutes	between 2-4	200-900[c]	Defined projects, set costs and budget using SF2 values, Budget = $ 4,000,000

a. The original free flow times differed for each link direction and were not applicable.
b. The original values were all 2 and this number of lanes could not lead to many asymmetric lane allocation scenarios for the network (with exception of new links with 4 lanes).
c. Demand values and reserve capacity of the resulting network were considered.

It is assumed that the original form of each network has its entire links as two-way streets. Construction costs have a linear relationship with the number of lanes added. The travel time on a link (t_{ij}) is a function of the flow on it, and can be described by Bureau of Public Roads (BPR) function, in which α is assumed 0.15 and β equal to 4. The functional form is indicated below where t^0_{ij} is the free flow travel time on arc (i, j).

$$t_{ij}(x_{ij}) = t^0_{ij}\left(1 + \alpha\left(\frac{x_{ij}}{c_{ij}}\right)^{\beta}\right) \tag{41}$$

5.2. Parameter setting

Parameter setting is one of the important steps of solving in meta-heuristics. It affects the quality of solutions obtained together with the runtime of the algorithm. To set the parameters of the three algorithms, we first reviewed parameter ranges in similar studies, and then conducted experiments for each parameter value. The combination of parameters for each algorithm was set such that the overall computational effort of algorithms for each problem becomes as close as possible together. Moreover, the same parameter settings were used for both DNDP1 and DNDP2. Table 5 shows the parameters for the three metaheuristics.

Table 5. Parameter settings for the algorithms

Algorithm	Main parameters	SA parameters
mHGA	Population size: 60 Number of generations: 300 Portion of solutions to apply SA: 1/3	Start temperature: 5 Reduction rate for small examples: 0.9 Reduction rate for other examples: 0.8 Stop temperature: 1
mESA	Population size: 60 Number of generations for small examples: 84×N Number of generations for other examples: 49×N	Start temperature: 10 Reduction rate: 0.95 Stop temperature: 1
mHABC	Population size: 15 Number of cycles for small examples: 4.5×N Number of cycles for other examples: 2.6×N Limit: 4 Number of employer bees and onlookers = 15	Start temperature: 10 Reduction rate: 0.92 Stop temperature: 1

The computational effort of each algorithm is a function of the number of generated solutions. The number of these solutions in turn, is dependent to the parameter setting of the algorithms. Thus, the approximate total computational effort can be expressed in terms of these parameters. For mHGA, the effort is roughly equal to $G \times (2/3 \times N \times itr + 2 \times N)$, in which *N* is the number of nodes and *itr* is the number of solutions generated by SA iterations. It is because crossover operation generates 2×*N* solutions; 1/3 of them were applied SA and *itr* SA iterations were applied on each of the selected solutions. The portion 1/3 is chosen to trade-off high computational effort and the solution quality. For mESA, the effort is about $G \times itr$. This is because *itr* numbers of solutions are generated by SA iterations for each of the *G* generations. For mHABC, the approximate computational effort can be formulated as $2 \times C \times P \times itr$, since *itr* numbers of solutions are generated by SA twice for each of *P* population solutions in *C* cycles. These numbers of solutions generated were applied as a guideline for setting the parameters of the algorithms.

By referring to the similar papers, we set the number of generations of mHGA and mESA and their population size. Carrying out some experiments also helped a lot in fixing the values of parameters. But the mHABC has a different algorithmic structure and its computational time is high. Also, there are few papers from which we can get help in this regard. Therefore, the size of population and the number of cycles differ from the other methods. Other values of parameters like the number of onlookers and scouts, the limit, population size, and the number of cycles has been set by carrying out enough experiments.

Two criteria are taken into account for setting the values of parameters of SA. At first, in the cooling procedure, the worst solution should be accepted with a reasonable probability, and then, the whole algorithm should have a rational computational effort. The SA parameters for mHGA are different for small, medium, and large test examples. This is motivated by the fact that computational effort for mHGA is significantly dependent on the value of *N* as explained earlier, while the number of generated solutions in mESA and mHABC are independent from the problem size.

5.3. Software and hardware

All algorithms were coded by Matlab version 7 and the tests were carried out on a laptop with a Core2Duo T7500 2.2GHz CPU, and a 2G RAM. Each algorithm was run for 5 times.

5.4. Performance Evaluation

To evaluate the performance of the algorithms, two methods are applied. Both of them are based on the assessment of Pareto-optimal sets:

- Comparison using the best values of each individual objective function: this can help to assess to what extent the algorithms can explore the solution space.

- Comparison using effectiveness measures of Pareto-optimal sets: This method compares the outputs of the algorithms using Pareto-optimal assessment measures.

Both comparisons are carried out in two ways; one by solving the smallest test example HF and comparing the results with those of three algorithms, and the other by comparing the results of the three algorithms directly with each other. The example HF was solved using an exact method. Actually the runtime of this method is very high even for this small example (more than 12 hours).

The specific characteristic of our bilevel NDP makes it difficult to apply the branch-and-bound method as an exact algorithm to solve them. The branch-and-bound method requires lower bound estimation throughout the solution procedure, while this estimation cannot be easily done for our bilevel NDPs, in which the lower level problem is a user-equilibrium assignment problem. It is because there is no-closed form equation to depict the relationship between the values of design variables on the user equilibrium flows. Although LeBlanc (1975) proposed a method to compute the lower bound of DNDPs with link addition projects, it cannot be used here to solve DNDP1 and DNDP2. It is because the lower bound calculation method proposed by LeBlanc is based on computing the system optimal flows which gives a lower bound for total travel time under user optimal flows, while the two problems in this paper use other objective functions rather than total travel time and this paper considers multiple objectives, not single objective.

The exact algorithm developed here to solve example HF is a two-part enumeration method:

First Part: This part finds the feasible combinations of expansion and construction projects in terms of the budget consideration. The output is the set of feasible strategic decisions.

Second Part: This part applies an enumeration algorithm for each feasible combination of strategic decisions. The variables in this part are only the tactical decisions. The main steps are:

- Repeat until all feasible combinations are examined:
    - Select a feasible combination of expansion and construction projects:
        - Examine all possible combinations of street orientation and lane allocation.
        - Discard the disconnected network solutions (i.e. infeasible ones).
        - Compute the objective function values for each feasible solution.
    - If the Pareto-optimal set is null, then build it using the obtained solutions. Otherwise, update the Pareto-optimal set using the obtained solutions.

This enumeration method is a branch and bound algorithm without lower bound calculation, and with only the infeasibility as the bounding rule. By filtering the budget-infeasible combinations of construction and expansion projects, the original solution space is transformed into a smaller or reduced solution space which does not include infeasible combinations of projects, and then the enumeration method only requires searching in the reduced solution space. Even by reducing the size of the solution space, the computational effort of HF remains rather high. No example other than HF could be solved, since the size of reduced solution space of even DNDP2 in the ND case equals 229,352,929,488, which is computationally prohibitive to be solved exactly.

Comparison of Individual Objective Function Values

In order to evaluate the capability of the algorithms in achieving optimum or near optimal values for each individual objective function, we compared the corresponding results of test example HF obtained by these algorithms. Based on the results from solving example HF by the exact method, the individual optimum values of the objective functions Z_1, Z_2 and Z_3 are respectively equal to 2.742, 1.0009, and 0.03879 for DNDP1 and to 2.742, 1.0009, and 0.00019 for DNDP2.

A comparison of the algorithms can be made by assessing the best values obtained for each of the objective functions among all runs. Table 6 depicts the averages of best objective function values obtained among all 5 runs. Bold numbers indicate the best values among three algorithms. From this table, it can be seen for example HF that mHGA got all the best values of the three objective functions for DNDP2, and the best values of Z_1 and Z_2 for DNDP1. Also mESA and mHABC obtained the best values of Z_3 for DNDP2.

A comparison of values of the three objective functions in all 7 examples (i.e., 21 values of objective functions) for each algorithm reveals that mHGA obtained 18 best values; mHABC reached 2 best and 14 second best values, and mESA gave 1 best and 9 second best values for DNDP1. For DNDP2, mHGA achieved 19 best and 1 second best values, mHABC reached 2 best and 13 second best values, and mESA provided 2 best and 8 second best values for DNDP2. It is observed that mHGA outperforms other algorithms in all three large test examples in terms of each objective. Therefore, mHGA is the preferred solution procedure in terms of the values of the objective functions obtained. The best values are shown in bold in the table.

Table 6. The best objective function values for both DNDP1 and DNDP2

Example	mHGA			mESA			mHABC		
	Z_1	Z_2	Z_3	Z_1	Z_2	Z_3	Z_1	Z_2	Z_3
DNDP1									
HF	**2.742**	**1.0009**	0.00076	2.730	1.0010	0.00048	2.705	1.0011	**0.00035**
ND	**1.158**	**1.0314**	0.56162	1.144	1.0372	0.55793	1.156	1.0345	**0.55682**
SF1	**0.634**	**1.1483**	**3.39290**	0.591	1.1924	7.72480	0.606	1.1894	7.16390
NA1	**1.866**	**1.0021**	57.86200	1.520	1.0046	**55.34100**	1.525	1.0057	55.52400
NA2	**2.309**	**1.0015**	**26.00100**	2.038	1.0028	26.00200	1.789	1.0038	26.00200
SF2	**0.292**	**2.0202**	**29.05500**	0.256	2.5030	57.86900	0.253	2.2860	54.26300
NA3	**3.155**	**1.0004**	**52.03200**	2.372	1.0010	52.08200	2.609	1.0010	52.04300
DNDP2									
HF	**2.742**	**1.0009**	**0.03879**	2.705	1.0013	**0.03879**	2.680	1.0013	**0.03879**
ND	**1.166**	**1.0346**	**0.56119**	1.161	1.0386	0.56574	1.156	1.0360	0.56276
SF1	**0.581**	**1.1876**	**5.19340**	0.544	1.2276	12.10900	0.545	1.1925	8.47130
NA1	**1.698**	**1.0025**	58.18800	1.539	1.0050	**54.40900**	1.509	1.0051	55.09700
NA2	**1.848**	**1.0030**	**26.00400**	1.764	1.0036	26.00500	1.775	1.0044	26.08400
SF2	**0.275**	**2.0459**	**35.01600**	0.233	2.7252	75.46500	0.233	2.5212	63.32200

NA3	**3.434**	**1.0003**	52.08700	2.708	1.0006	52.08800	2.761	1.0006	**52.04400**

Table 7 summarizes the best values of each objective function obtained from all algorithms and gives the percentage of deviation of the best values obtained by DNDP2 with respect to the corresponding values obtained by DNDP1. The comparison of the best values of the individual objective functions for each test example between DNDP1 and DNDP2 reveals that DNDP1 leads to better objective function values than DNDP2. As can be seen from the table, DNDP1 has higher or equal values of demand matrix multipliers (i.e., the objective function Z_1) in 5 out of 7 examples, lower or equal values of the objective function Z_2 in 6 out of 7 examples, and lower or equal values of the objective function Z_3 in 6 out of 7 examples. This explains that more flexibility in lane allocations in DNDP1 provides higher reserve capacity and lower travel time. Therefore, allowing for two-way streets with unequal number of lanes at each direction at least for some of streets can improve road network performances.

Table 7. Comparison of the results between DNDP1 and DNDP2

Example	**DNDP1**			**DNDP2**			**Percentage of deviation of best values with respect to DNDP1**		
	Z_1	***Z_2***	***Z_3***	***Z_1***	***Z_2***	***Z_3***	***ΔZ_1(%)***	***ΔZ_2(%)***	***ΔZ_3(%)***
HF	2.742	1.0009	0.00035	2.742	1.0009	0.03879	0.00	0.00	10988.37
ND	1.158	1.0314	0.55682	1.166	1.0346	0.56119	0.67	0.31	0.78
SF1	0.634	1.1483	3.39290	0.581	1.1876	5.19340	-8.38	3.42	53.07
NA1	1.866	1.0021	55.34100	1.698	1.0025	54.40900	-8.96	0.04	-1.68
NA2	2.309	1.0015	26.00100	1.848	1.0030	26.00400	-19.96	0.15	0.01
SF2	0.292	2.0202	29.05500	0.275	2.0459	35.01600	-5.88	1.27	20.52
NA3	3.155	1.0004	52.03200	3.434	1.0003	52.04400	8.87	-0.01	0.02

Effectiveness Measures

In this section we propose two measures that are used for investigating the quality of the Pareto-optimal sets for each algorithm. M_1 is the first measure that takes the size of the Pareto-optimal set into consideration. The second measure, named M_2 which was proposed by Miandoabchi *et al.* (2012), identifies the overall performance of one algorithm, compared to other ones.

M_2 is based upon the concept of set coverage measure in Zitzler *et al.* (2000). They introduced a measure that compares the fraction of a Pareto-optimal set that dominates another Pareto-optimal set. In the following, the mathematical form of this measure is given:

$$C(X_i, X_j) = \frac{\left|\{a_j \in X_j ; \exists a_i \succcurlyeq a_j\}\right|}{\left|X_j\right|} \tag{42}$$

where the domination or equality of solution a_i to a_j is represented by $a_i \succcurlyeq a_j$. Equation (42) calculates the fraction of solutions in set X_j that is dominated by or equal to at least one solution in X_i. In other words, the fraction of the solutions in set X_j that is covered by set X_i is calculated. According to this definition, $C(X_i, X_j)$ is not necessarily equal to 1- $C(X_j, X_i)$.

Based on this set coverage concept, we can define a measure to indicate the overall strength of an algorithm with respect to other algorithms. This measure should have a larger value when the algorithm can obtain more non-dominated and less dominated solutions than other algorithms. In other words, the more the fraction of non-dominated solutions and the less the fraction of dominated solutions, the higher should be the value of the measure, and vice versa. Therefore, for developing *M₂*, we sum up the *C* values that represent the fraction of solutions which dominate the solutions of other algorithms and then subtract all *C* values which represent the fraction of solutions that is dominated by the solutions of other algorithms. The proposed formula for M_2 is given in (43).

$$M_2(X_i) = \sum_{j \neq i} C(X_i, X_j) - \sum_{j \neq i} C(X_j, X_i) \qquad (43)$$

In tables 8 and 9, the average values of the measures and the average runtimes for solving DNDP1 and DNDP2 are summarized. Regarding to the values of the measure M_1, out of 14 values for both problem variations, mHGA has the first ranking with 7 best and 4 second best values, mESA has the second ranking with 5 best and 4 second best values, and mHABC has the third ranking with 4 best and 7 second best values. The best values in tables 8-9 are indicated in bold.

Table 8. Summary of computational results for DNDP1

Example	mHGA			mESA			mHABC		
	M_1	M_2	*Runtime**	M_1	M_2	*Runtime**	M_1	M_2	*Runtime**
HF	24	**0.81**	**448**	**35**	-0.37	558	**35**	-0.44	564
ND	10	**1.22**	**1,104**	**13**	-1.03	1,618	11	-0.19	1,633
SF1	**40**	**1.93**	**9,111**	12	-0.83	9,553	19	-1.10	9,533
NA1	6	**1.43**	**4,652**	**10**	-1.02	4,811	8	-0.42	4,880
NA2	10	**1.88**	**5,159**	14	-0.70	5,232	**15**	-1.17	5,362
SF2	**36**	**2.00**	**27,919**	8	-1.64	29,232	8	-0.36	29,349
NA3	**17**	**1.79**	**19,575**	15	-1.62	25,294	11	-0.16	25,321

* Average runtime (in seconds)

Table 9. Summary of computational results for DNDP2

Example	mHGA			mESA			mHABC		
	M_1	M_2	*Runtime**	M_1	M_2	*Runtime**	M_1	M_2	*Runtime**
HF	19	**1.20**	**415**	**30**	-0.28	567	**30**	-0.92	573
ND	12	**1.29**	**1,149**	**13**	-1.07	1,623	11	-0.22	1,642
SF1	**31**	**1.26**	**8,987**	14	-1.00	9,575	21	-0.26	9,529
NA1	6	**1.00**	**4,668**	5	-0.40	5,191	**11**	-0.60	5,165
NA2	**21**	**0.74**	**4,512**	15	-0.16	5,382	14	-0.57	5,470
SF2	**29**	**1.98**	**26,020**	10	-1.35	29,251	14	-0.63	29,479
NA3	**17**	**1.31**	**19,242**	9	-0.94	26,171	13	-0.36	26,131

* Average runtime (in seconds)

In order to obtain rankings of algorithms with respect to M_1 and M_2, the following steps were taken for DNDP1 and DNDP2:

1. Ranking the three algorithms for each measure, each test example and each problem variation.
2. Counting the total number of 1^{st}, 2^{nd} and 3^{rd} rankings of each algorithm for each measure among the test examples and problem variations.
3. Computing the score of each algorithm in each measure by combining ranking its counts using the weighted sum method: (No. of 1^{st} rankings)*1 + (No. of 2^{nd} rankings)*0.5 + (No.

of 3rd rankings)*0.25.

Table 10 summarizes the computations to obtain algorithm scores.

Table 10. Algorithm scores for each measure

Algorithm / Criteria	mHGA		mESA		mHABC	
	M_1	M_2	M_1	M_2	M_1	M_2
No. of 1st rankings	7	14	5	0	4	0
No. of 2nd rankings	4	0	4	6	7	8
No. of 3rd rankings	3	0	5	8	3	6
Scores	9.8	14	8.25	5	8.3	5.5

Based on the scores obtained, we can conclude that mHGA outperforms other algorithms in all three large examples, and thus it is the most preferable solution procedure, especially for large examples.

The runtimes of the algorithms for each test example are at the same range, as the total computational efforts are controlled to be similar. The following shows the rank of the solution algorithms under each criterion.

- M_1: mHGA > mHABC > mESA
- M_2: mHGA > mHABC > mESA
- best objective function values: mHGA > mHABC > mESA

Both the M_1 and M_2 and also the best values of individual objective functions suggest that mHGA performs the best.

Figures 7 and 8 illustrate the average runtimes of all the algorithms for each test example for DNDP1 and DNDP2. As can be seen from these figures, the runtimes for test examples SF1 and SF2 are slightly higher than their neighbor test examples with similar network sizes. This can be explained by the existence of the higher numbers of OD pairs in SF1 and SF2, which require higher computation time to take into account the interactions of flows from different OD pairs. This indicates the significant effect of the number of OD pairs on the computation time required.

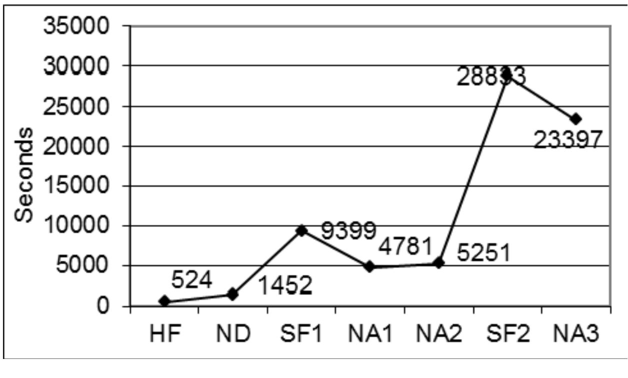

Figure 7. Average runtimes for DNDP1

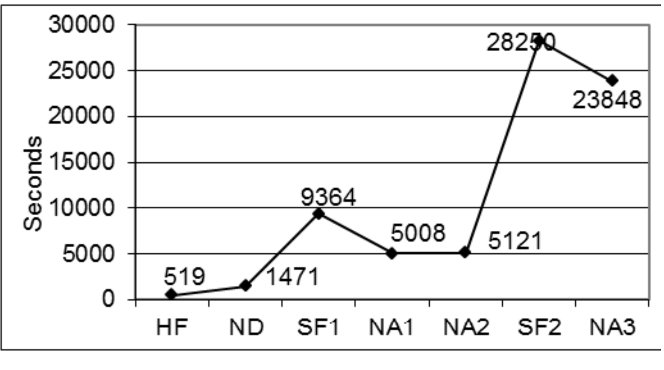

Figure 8. Average runtimes for DNDP2

Performance in Achieving the True Pareto-optimal Set

In order to evaluate the degree of closeness of the obtained Pareto-optimal solution set by each algorithm to the optimum set, the exact solution of example HF was used. The non-dominated solutions among the union of Pareto-optimal sets among 5 runs for each algorithm were found, and the percentage of true Pareto-optimal solutions achieved by each algorithm was calculated. The results are given in Table 11. It can be observed that mHGA can obtain the largest percentage.

Table 11. Percentages of true Pareto-optimal solutions found for HF

		DNDP1	***DNDP2***
Percentages	**mHGA**	47.9 %	100 %
	mESA	22.9 %	47.6 %
	mHABC	25 %	38.1 %
Size of true Pareto-optimal set		48	21
Size of original solution space		86,400,000	839,808
Size of reduced solution space evaluated by the exact method		3,294,507	70,399
Runtime of the exact method		68,074	1,042

The sizes of the original and reduced solution spaces for the two problems and the runtimes required are also shown in Table 11. The original solution space size for DNDP2 is calculated using the expression $\prod_{\mathrm{l}\in\mathrm{L}}\left(3\left(1+\mathrm{y}_{\mathrm{l}}^{\max}\right)\right).4^{|\mathrm{L}'|}$. It counts 3 possibilities for an existing link's direction (2 possibilities for a one-way street and 1 possibility for a two-way street) multiplied by 1 + $y_l^{\max}$ possibilities for its expansion (Here, $y_l^{\max}$ is either 0 or 1), and 4 possibilities for candidate link's direction (1 possibility for the link not being built and 3 possibilities for the link to be built). Since there are 5 non-expandable links, 3 expandable links with maximum 1 lane addition on each side, and 2 constructible links for HF, the expression will be $(3 \times (1 + 0))^5 \times (3 \times (1 + 1))^3 \times 4^2$ which gives 839,808.

For DNDP1, the expression $\prod_{\mathrm{l}\in\mathrm{L}}\sum_{\mathrm{y_l}=0}^{\mathrm{y_l}=1}(\mathrm{K_l}+2\mathrm{y_l}+1).\prod_{\mathrm{l'}\in\mathrm{L'}}(\mathrm{K_{l'}}+2)$ is used. The first product term counts all lane allocation possibilities for existing links. $K_l + 2y_l$ is the sum of the number of existing lanes and the number of lanes to be added to the existing link. This leads to $K_l + 2y_l + 1$ possibilities of asymmetric lane allocations, after including 1 possibility for not expanding it. If a link is non-expandable (i.e. $y_l = 0$), then lane allocation possibilities becomes $K_l + 1$. If a link is expandable, then possible lane allocations must be counted for both expansion ($y_l = 1$) and no-expansion ($y_l = 0$) cases. Thus, all possibilities for expansion or no-expansion must be summed up. The second product term of the expression counts $K_{l'} + 1$ possibilities for asymmetric lane allocations of a constructed new link, plus 1 possibility for not constructing it.

For HF, there are 2 construction projects (with 4 lanes), 3 expansion projects (with 2 lane additions) for two 2-lane links and one 4-lane link. All other 5 links have 4 lanes. Thus, for 2 constructible links, the expression will be $(4 + 2)^2$; for two expandable links with existing 2 lanes, the expression will be $[(2 + 2 \times 0 + 1) + (2 + 2 \times 1 + 1)]^2$; for one expandable link with existing 4 lanes, the expression will be [(4 + 2 × 0 + 1) + (4 + 2 × 1 + 1)], and for the remaining 5 non-expandable links, the expression will be $[(4 + 2 \times 0 + 1)]^5$. Multiplying all the above values will give 86,400,000 as the size of the original space.

For the reduced solution space in DNDP1 or DNDP2, after creating the feasible combinations of projects, the above expression was used to determine the possibilities for each combination. The possibilities of each combination were summed up to obtain the reduced solution space size.

From the table, we can see that the reduced solution spaces for DNDP1 and DNDP2 are respectively about one-twentieth and one-tenth of their individual original solution space. Moreover, the computation time of the exact method for solving DNDP2 is only 2% of that for DNDP1 because the reduced solution space in DNDP2 is also about 2% of that for DNDP1.

Sensitivity Analysis of Parameter Settings

To investigate the sensitivity of the computational results to the parameter settings of the algorithms, a series of experiments were conducted. Since the total computational effort of the algorithms must be kept fixed, experiments could not be performed by changing one parameter at a time (except for population size in mHGA and mESA). Thus, various parameter setting strategies were generated by changing the values of two parameters at one time, keeping the current values of total computational effort unchanged. The total number of temperature changes in each SA iteration, was considered as one of the parameters, which can be adjusted using the reduction rate. The start and stop temperatures were set fixed. In this way, all algorithms had the number of generations/cycles (*G*/*C*), population size (*P*), and the number of embedded SA iterations (*itr*) as their main parameters. The following experiments were conducted:

- mHGA and mESA: experiment 1 for (*G*, *itr*), experiment 2 for *P*
- mHABC: experiment 1 for (*C*, *P*), experiment 2 for (*C*, *itr*), and experiment 3 for (*P*, *itr*)

All experiments were run for problem variation DNDP1 once for each test problem. For each experiment, 6 scenarios were generated, covering a range of lower and higher values relative to the current parameter values. The results of each scenario were compared against the previous results of five runs of the other two algorithms, and the average values were reported. Table 12compares the experiments of the three algorithms using measures M_1 and M_2 for each test problem. The best values are indicated in bold.

To find out if the performance of the algorithms is sensitive to the parameter settings, first the ranking of each algorithm (i.e., 1st, 2nd, or 3rd) for each measure in each test example and for each scenario was obtained. Then the numbers of 1st, 2nd and 3rd rankings of each algorithm for each measure in each test problem among each complete set of experiments were counted. Each set of three ranking counts were combined using the weighted sum method described in the main computational tests. The obtained results are shown in table 13. The best performances for each scenario are shown in bold. To analysis the overall sensitivity of each algorithm, the scores were summed up for all test problems. These values can be used to compare the rankings of the algorithms with the ones reported in the main computational tests.

Table 13. Algorithm scores from sensitivity analysis tests

Sensitivity Analysis Alg.	Example	M_1			M_2		
		mHGA	mESA	mHABC	mHGA	mESA	mHABC
mHGA	HF	6	**12**	**12**	**11.25**	6	3.75
	ND	**10**	9	4.25	**11.25**	3.25	6.5
	SF1	**12**	3.25	6.5	**12**	6	3
	NA1	7.25	**9.5**	4.25	6.25	4.25	**10.5**
	NA2	7.5	4.5	**9**	**12**	6	3
	SF2	**11**	7	7	**11.5**	3	6.5
	NA3	7.8	**10**	4	**12**	3	6
Total		***61.55***	*55.3*	*47*	***76.25***	*31.5*	*39.25*
mESA	HF	3.5	**9.5**	8	**12**	3.75	5.25
	ND	4.5	**9**	**9**	**12**	3.5	5.5
	SF1	**12**	3	6	**12**	5.5	3.5
	NA1	4.25	**10.5**	8.5	**12**	3	6
	NA2	3.5	**10.5**	7	**12**	3.5	5.5
	SF2	**12**	4.25	4.75	**12**	4	5
	NA3	**11**	5	5	**12**	3.5	5.5
Total		*50.75*	***52***	*48.25*	***84***	*26.8*	*36.25*
mHABC	HF	4.75	13	**13.75**	**18**	7	6.5
	ND	5.5	**13**	**13**	**18**	5.5	8
	SF1	**18**	5.75	7.75	**16.5**	6	9
	NA1	5.5	**15**	11	**18**	8.5	5
	NA2	5	12	**14.5**	**18**	7.5	6
	SF2	**18**	7.75	5.75	**18**	8.75	4.75
	NA3	**15.5**	7.75	8.25	**18**	5	8.5
Total		*72.25*	***74.3***	*74*	***124.5***	*48.3*	*47.75*

According to the total scores in table 13, the ranking of each algorithm for each measure will be as in table 14. The obtained rankings are compared to the original ones (M1: mHGA > mHABC > mESA; M2: mHGA > mHABC > mESA), and the result for each algorithm is explained in the last column.

Table 14. Algorithm rankings according to sensitivity analysis test

Measure	Algorithm under Sensitivity Analysis	Ranking Obtained after Sensitivity analysis	Conclusion
M_1	mHGA	mHGA > mESA > mHABC	Performance of mHGA is not sensitive to its parameter settings
	mESA	mESA > mHGA > mHABC	Performance of mESA is sensitive to its parameter settings
	mHABC	mHABC > mESA > mHGA	Performance of mHABC is sensitive to its parameter settings
M_2	mHGA	mHGA > mHABC > mESA	Performances of mHGA and mESA are not, but the performance of mHABC is sensitive to its parameter settings
	mESA	mHGA > mHABC > mESA	
	mHABC	mHGA > mESA > mHABC	

The above computational results lead us to the conclusion that the comparative performance of mHGA in both measures and mESA in measure M_2 are not sensitive to the changes in the algorithms' parameters. However, the performance of mHABC in both measures and mESA in measure M_2 is sensitive to the algorithms' parameters. Overall, it can be concluded that the performance of mHGA is the least affected by changes in parameter settings and thus it is the most reliable and desirable algorithm to solve the problems.

Table 12. Summary of computational results for sensitivity analysis

Problem	mHGA											mESA										mHABC									
	Experiment	*Scenario*	*Parameters*			*mHGA*		*mESA*		*mHABC*		*Experiment*	*Parameters*			*mHGA*		*mESA*		*mHABC*		*Experiment*	*Parameters*			*mHGA*		*mESA*		*mHABC*	
			G	*P*	*itr*	M_1	M_2	M_1	M_2	M_1	M_2		*G*	*P*	*itr*	M_1	M_2	M_1	M_2	M_1	M_2		*C*	*P*	*itr*	M_1	M_2	M_1	M_2	M_1	M_2
HF	No. 1 *G* and *itr*	1	150	60	35	22	**0.59**	**35**	-0.27	**35**	-0.33	No. 1 *G* and *itr*	207	60	110	24	**1.01**	**56**	-0.89	35	-0.12	No. 1 *C* and *P*	12	34	28	24	**0.83**	35	-0.33	**37**	-0.50
		2	200	60	25	33	**0.64**	**35**	-0.25	**35**	-0.39		307	60	74	24	**0.88**	**43**	-0.59	35	-0.29		17	24	28	24	**0.84**	**35**	-0.70	27	-0.13
		3	250	60	20	26	**0.52**	**35**	-0.19	**35**	-0.34		407	60	56	24	**0.82**	23	-0.41	**35**	-0.40		22	19	28	24	**1.06**	35	-0.10	**57**	-0.96
		4	350	60	13	25	**0.83**	**35**	-0.36	**35**	-0.48		607	60	38	24	**0.68**	33	-0.21	**35**	-0.48		32	13	28	24	**0.81**	**35**	-0.54	25	-0.28
		5	400	60	11	25	**0.78**	**35**	-0.36	**35**	-0.42		707	60	32	24	**0.77**	26	-0.29	**35**	-0.48		37	11	28	24	**0.76**	**35**	-0.60	32	-0.16
		6	450	60	10	32	**0.52**	**35**	-0.24	**35**	-0.28		807	60	28	24	**0.92**	**38**	-0.59	35	-0.34		42	10	28	24	**0.84**	35	-0.25	**45**	-0.59
	No. 2 *P*	7	300	30	16	27	-0.19	**35**	**0.08**	**35**	0.11	No. 2 *P*	507	30	45	24	**0.83**	**36**	-0.55	35	-0.28	No. 2 *C* and *itr*	12	15	63	24	**0.75**	**35**	-0.54	29	-0.21
		8	300	40	16	24	**0.24**	**35**	-0.09	**35**	-0.15		507	40	45	24	**0.55**	24	0.06	**35**	-0.61		17	15	45	24	**0.83**	**35**	-0.42	31	-0.41
		9	300	50	16	28	**0.36**	**35**	-0.12	**35**	-0.24		507	50	45	24	**1.04**	**57**	-0.90	35	-0.14		22	15	35	24	**1.05**	35	-0.07	**43**	-0.98
		10	300	70	16	26	**0.81**	**35**	-0.34	**35**	-0.47		507	70	45	24	**0.80**	**37**	-0.52	35	-0.28		32	15	24	24	**0.49**	**35**	-0.76	31	0.27
		11	300	80	16	26	**0.54**	**35**	-0.18	**35**	-0.36		507	80	45	24	**1.01**	**60**	-0.71	35	-0.30		37	15	21	24	**0.82**	35	-0.40	**39**	-0.42
		12	300	90	16	29	**0.41**	**35**	-0.14	**35**	-0.27		507	90	45	24	**1.05**	**58**	-0.80	35	-0.25		42	15	18	24	**1.06**	35	-0.43	**62**	-0.64
	–	13	–									–	–									No. 3 *P* and *itr*	27	5	84	24	**1.33**	35	0.36	**37**	-1.68
		14																					27	10	42	24	**1.09**	35	-0.18	**59**	-0.91
		15																					27	20	21	24	**0.68**	**35**	-0.80	32	0.12
		16																					27	25	17	24	**0.88**	35	-0.41	**61**	-0.47
		17																					27	30	14	24	**0.96**	35	-0.14	**58**	-0.81
		18																					27	35	12	24	**0.65**	**35**	-0.70	22	0.04
ND	No. 1 *G* and *itr*	1	150	60	35	**13**	**1.23**	**13**	-1.08	11	-0.15	No. 1 *G* and *itr*	798	60	62	10	**1.35**	**16**	-1.44	11	0.09	No. 1 *C* and *P*	12	34	28	10	**1.46**	13	-0.51	**19**	-0.95
		2	200	60	25	7	**1.30**	**13**	-1.13	11	-0.16		898	60	55	10	**0.91**	10	-0.53	**11**	-0.38		17	24	28	10	**1.16**	13	-1.29	**17**	0.13
		3	250	60	20	**15**	**1.43**	13	-1.10	11	-0.32		998	60	50	10	**1.30**	10	-1.20	**11**	-0.10		22	19	28	10	**1.50**	**13**	-0.80	8	-0.71
		4	350	60	13	**17**	**1.34**	13	-1.05	11	-0.29		1198	60	41	10	**1.31**	**12**	-1.04	11	-0.27		32	13	28	10	**1.40**	13	-0.89	**16**	-0.51
		5	400	60	11	**16**	**1.16**	13	-0.89	11	-0.27		1298	60	38	10	**1.24**	**16**	-0.74	11	-0.50		37	11	28	10	**1.47**	**13**	-0.75	11	-0.72
		6	450	60	10	**34**	-1.15	13	0.03	11	**1.13**		1398	60	35	10	**1.36**	**11**	-1.33	**11**	-0.03		42	10	28	10	**1.30**	13	-1.20	**16**	-0.10
	No. 2 *P*	7	300	30	16	12	**1.28**	**13**	-1.04	11	-0.24	No. 2 *P*	1098	30	45	10	**1.37**	**23**	-1.55	11	0.18	No. 2 *C* and *itr*	12	15	37	10	**1.04**	13	-1.38	**18**	0.34
		8	300	40	16	**14**	**0.41**	13	-0.76	11	0.34		1098	40	45	10	**0.91**	10	0.01	**11**	-0.92		17	15	34	10	**0.79**	**13**	-1.18	10	0.39
		9	300	50	16	10	**1.38**	**13**	-1.06	11	-0.32		1098	50	45	10	**1.22**	8	-0.99	**11**	-0.23		22	15	30	10	**1.03**	13	-1.23	**20**	0.21
		10	300	70	16	**13**	**0.98**	**13**	-0.87	11	-0.12		1098	70	45	10	**1.29**	**16**	-1.14	11	-0.15		32	15	26	10	**1.13**	**13**	-1.08	12	-0.05
		11	300	80	16	**13**	**0.88**	**13**	-0.85	11	-0.02		1098	80	45	10	**1.00**	8	-0.33	**11**	-0.67		37	15	24	10	**1.60**	**13**	-0.68	12	-0.92
		12	300	90	16	**14**	**1.33**	13	-1.05	11	-0.28		1098	90	45	10	**1.16**	**13**	-0.88	11	-0.28		42	15	22	10	**1.37**	13	-1.11	**20**	-0.26
	–	13	–									–	–									No. 3 *P* and *itr*	27	5	84	10	**1.24**	13	-0.54	**23**	-0.70
		14																					27	10	42	10	**1.56**	13	-0.20	**20**	-1.36
		15																					27	20	21	10	**1.08**	13	-1.22	**21**	0.14
		16																					27	25	17	10	**0.76**	**13**	-1.34	10	0.58
		17																					27	30	14	10	**1.47**	**13**	-0.95	8	-0.52
		18																					27	35	12	10	**1.17**	**13**	-1.29	12	0.12
SF1	No. 1 *G* and *itr*	1	150	60	35	**19**	**1.85**	12	-0.82	**19**	-1.04	No. 1 *G* and *itr*	582	60	91	**40**	**1.88**	9	-0.96	19	-0.92	No. 1 *C* and *P*	33	29	28	**40**	**1.45**	12	-1.56	10	0.11
		2	200	60	25	**49**	**1.95**	12	-0.85	19	-1.10		782	60	68	**40**	**1.33**	8	0.29	19	-1.62		43	22	28	**40**	**1.66**	12	-1.61	15	-0.05
		3	250	60	20	**37**	**1.93**	12	-0.85	19	-1.08		982	60	54	**40**	**1.48**	8	-0.08	19	-1.40		53	18	28	**40**	**1.82**	12	-0.93	16	-0.89
		4	350	60	13	**31**	**1.98**	12	-0.87	19	-1.11		1382	60	38	**40**	**1.01**	2	0.46	19	-1.48		73	13	28	**40**	**1.89**	12	-0.82	19	-1.07
		5	400	60	11	**35**	**1.94**	12	-0.85	19	-1.09		1582	60	34	**40**	**1.79**	15	-0.64	19	-1.15		83	11	28	**40**	**1.71**	12	-1.56	20	-0.15
		6	450	60	10	**37**	**1.89**	12	-0.83	19	-1.05		1782	60	30	**40**	**1.91**	10	-1.46	19	-0.45		93	10	28	**40**	**1.94**	12	-0.65	21	-1.28

Problem	mHGA											mESA										mHABC									
	Experiment	*Scenario*	*Parameters*			*mHGA*		*mESA*		*mHABC*		*Experiment*	*Parameters*			*mHGA*		*mESA*		*mHABC*		*Experiment*	*Parameters*			*mHGA*		*mESA*		*mHABC*	
			G	*P*	*itr*	M_1	M_2	M_1	M_2	M_1	M_2		*G*	*P*	*itr*	M_1	M_2	M_1	M_2	M_1	M_2		*C*	*P*	*itr*	M_1	M_2	M_1	M_2	M_1	M_2
	No. 2 *P*	7	300	30	16	**31**	**1.98**	12	-0.87	19	-1.11	No. 2 *P*	1182	30	45	**40**	**1.80**	12	-0.48	19	-1.32	No. 2 *C* and *itr*	33	15	54	**40**	**1.95**	12	-0.37	22	-1.59
		8	300	40	16	**32**	**1.83**	12	-0.80	19	-1.04		1182	40	45	**40**	**1.72**	8	-0.80	19	-0.92		43	15	41	**40**	**1.96**	12	-0.48	11	-1.48
		9	300	50	16	**30**	**1.94**	12	-0.85	19	-1.09		1182	50	45	**40**	**1.66**	9	-0.46	19	-1.20		53	15	33	**40**	**1.52**	12	-1.60	17	0.08
		10	300	70	16	**44**	**1.96**	12	-0.87	19	-1.09		1182	70	45	**40**	**1.90**	15	-0.87	19	-1.04		73	15	24	**40**	**1.10**	12	-1.80	7	0.70
		11	300	80	16	**32**	**1.83**	12	-0.80	19	-1.04		1182	80	45	**40**	**1.49**	13	-0.01	19	-1.48		83	15	21	**40**	0.70	12	-1.92	15	**1.23**
		12	300	90	16	**32**	**1.94**	12	-0.85	19	-1.09		1182	90	45	**40**	**1.86**	7	-0.71	19	-1.15		93	15	19	**40**	**1.57**	12	-1.51	9	-0.06
	–	13	–									–	–									No. 3 *P* and *itr*	63	5	84	**40**	**1.70**	12	-1.49	16	-0.22
		14																					63	10	42	**40**	**1.96**	12	-0.62	10	-1.34
		15																					63	20	21	**40**	**1.91**	12	-0.67	17	-1.24
		16																					63	25	17	**40**	0.56	12	-1.90	16	**1.34**
		17																					63	30	14	**40**	0.92	12	-1.88	16	**0.97**
		18																					63	35	12	**40**	**1.81**	12	-1.00	16	-0.80
NA1	No. 1 *G* and *itr*	1	150	60	19	**12**	-1.25	10	0.28	8	**0.97**	No. 1 *G* and *itr*	378	60	116	6	**1.26**	**11**	-1.39	8	0.13	No. 1 *C* and *P*	37	21	28	6	**1.33**	**10**	-0.57	9	-0.77
		2	200	60	13	1	**0.82**	**10**	-0.76	8	-0.07		578	60	76	6	**1.50**	5	-1.57	**8**	0.07		42	19	28	6	**1.55**	**10**	-0.28	6	-1.27
		3	250	60	10	**14**	0.31	10	-0.63	8	**0.32**		778	60	57	6	**1.43**	**9**	-1.62	8	0.18		47	17	28	6	**1.54**	10	-0.38	**11**	-1.16
		4	350	60	6	2	**0.72**	**10**	-0.73	8	0.01		1178	60	37	6	**1.60**	**10**	-1.82	8	0.22		57	14	28	6	**1.63**	**10**	-0.25	**10**	-1.38
		5	400	60	5	5	**0.79**	**10**	-0.84	8	0.05		1378	60	32	6	**1.41**	**13**	-1.49	8	0.08		62	13	28	6	**1.57**	10	-0.05	**11**	-1.52
		6	450	60	4	**17**	-1.59	10	0.46	8	**1.13**		1578	60	28	6	**1.62**	**12**	-1.85	8	0.23		67	12	28	6	**1.59**	**10**	-0.39	7	-1.20
	No. 2 *P*	7	300	30	8	10	-0.74	**10**	-0.08	8	**0.81**	No. 2 *P*	978	30	45	6	**1.57**	**14**	-1.73	8	0.16	No. 2 *C* and *itr*	22	15	67	6	**1.48**	**10**	-0.30	7	-1.17
		8	300	40	8	**12**	0.21	10	-0.59	8	**0.39**		978	40	45	6	**1.52**	**10**	-1.72	8	0.20		32	15	46	6	**1.64**	10	-0.26	**11**	-1.38
		9	300	50	8	**18**	-0.87	10	0.10	8	**0.77**		978	50	45	6	**1.30**	7	-1.25	**8**	-0.05		42	15	35	6	**1.23**	**10**	-0.91	2	-0.32
		10	300	70	8	2	-0.16	**10**	-0.44	8	**0.59**		978	70	45	6	**1.53**	**12**	-1.65	8	0.12		62	15	24	6	**1.52**	10	-0.27	**14**	-1.25
		11	300	80	8	8	-0.54	**10**	-0.03	8	**0.57**		978	80	45	6	**1.49**	**18**	-1.63	8	0.14		72	15	20	6	**1.55**	**10**	-0.37	6	-1.18
		12	300	90	8	1	0.01	**10**	-0.36	8	**0.35**		978	90	45	6	**1.50**	5	-1.33	**8**	-0.17		82	15	18	6	**1.63**	10	0.03	**20**	-1.66
	–	13	–									–	–									No. 3 *P* and *itr*	52	5	85	6	**1.64**	10	-0.14	**11**	-1.50
		14																					52	10	42	6	**1.44**	**10**	-0.19	9	-1.25
		15																					52	20	21	6	**1.58**	**10**	-0.46	7	-1.11
		16																					52	25	17	6	**1.52**	**10**	-0.55	5	-0.96
		17																					52	30	14	6	**1.22**	**10**	-0.85	2	-0.37
		18																					52	35	12	6	**1.40**	**10**	-0.35	3	-1.05
NA2	No. 1 *G* and *itr*	1	150	60	19	2	**0.71**	14	-0.13	**15**	-0.58	No. 1 *G* and *itr*	475	60	102	10	**1.78**	7	-0.45	**15**	-1.32	No. 1 *C* and *P*	28	31	28	10	**1.93**	**14**	-1.03	13	-0.90
		2	200	60	14	**16**	**1.80**	14	-0.66	15	-1.14		675	60	72	10	**1.93**	**20**	-1.56	15	-0.37		38	23	28	10	**1.95**	14	-0.51	**17**	-1.44
		3	250	60	10	**17**	**1.82**	14	-0.68	15	-1.14		875	60	55	10	**1.93**	**18**	-1.43	15	-0.50		48	18	28	10	**1.83**	14	-0.59	**21**	-1.24
		4	350	60	6	3	**1.72**	14	-0.60	**15**	-1.12		1275	60	38	10	**1.93**	**20**	-1.05	15	-0.88		68	13	28	10	**1.90**	14	-0.80	**16**	-1.10
		5	400	60	5	6	**1.71**	14	-0.61	**15**	-1.10		1475	60	33	10	**1.91**	**23**	-1.41	15	-0.50		78	11	28	10	**1.95**	14	-0.95	**15**	-1.00
		6	450	60	4	**18**	**1.82**	14	-0.68	15	-1.14		1675	60	29	10	**1.93**	**21**	-1.53	15	-0.40		88	10	28	10	**1.68**	14	-1.32	**17**	-0.36
	No. 2 *P*	7	300	30	8	6	**1.89**	14	-0.74	**15**	-1.15	No. 2 *P*	1075	30	45	10	**1.93**	6	-1.07	**15**	-0.86	No. 2 *C* and *itr*	28	15	58	10	**1.95**	14	-0.38	**20**	-1.57
		8	300	40	8	**27**	**1.81**	14	-0.69	15	-1.12		1075	40	45	10	**1.87**	**15**	-1.09	**15**	-0.77		38	15	43	10	**1.95**	14	-0.55	**18**	-1.40
		9	300	50	8	12	**1.76**	14	-0.62	**15**	-1.14		1075	50	45	10	**1.93**	**15**	-0.98	**15**	-0.94		48	15	34	10	**1.92**	14	-0.65	**15**	-1.27
		10	300	70	8	**30**	**1.94**	14	-0.73	15	-1.21		1075	70	45	10	**1.93**	**17**	-1.33	15	-0.60		68	15	24	10	**1.87**	14	-0.59	**20**	-1.28
		11	300	80	8	**16**	**1.87**	14	-0.72	15	-1.15		1075	80	45	10	**1.93**	**23**	-1.41	15	-0.52		78	15	21	10	**1.93**	14	-1.52	**27**	-0.41
		12	300	90	8	10	**1.92**	14	-0.74	**15**	-1.18		1075	90	45	10	**1.93**	**15**	-0.86	**15**	-1.07		88	15	18	10	**1.80**	**14**	-1.39	10	-0.41
	–	13	–									–	–									No. 3 *P* and *itr*	58	5	83	10	**1.95**	**14**	0.05	12	-2.00
		14																					58	10	42	10	**1.95**	**14**	-0.74	9	-1.21

Problem	**mHGA**											**mESA**										**mHABC**									
	Experiment	***Scenario***	***Parameters***			***mHGA***		***mESA***		***mHABC***		***Experiment***	***Parameters***			***mHGA***		***mESA***		***mHABC***		***Experiment***	***Parameters***			***mHGA***		***mESA***		***mHABC***	
			G	***P***	***itr***	M_1	M_2	M_1	M_2	M_1	M_2		***G***	***P***	***itr***	M_1	M_2	M_1	M_2	M_1	M_2		***C***	***P***	***itr***	M_1	M_2	M_1	M_2	M_1	M_2
		15																					58	20	21	10	**1.87**	14	-0.50	**16**	-1.37
		16																					58	25	17	10	**1.95**	**14**	-1.32	11	-0.63
		17																					58	30	14	10	**1.92**	14	-0.93	**21**	-0.99
		18																					58	35	12	10	**1.86**	**14**	-1.01	12	-0.85
SF2	No. 1 *G* and *itr*	1	150	60	19	**25**	**0.98**	8	-1.33	8	0.35	No. 1 *G* and *itr*	573	60	92	**36**	**2.00**	5	-1.85	8	-0.15	No. 1 *C* and *P*	48	20	28	**36**	**2.00**	8	-0.06	10	-1.94
		2	200	60	13	**22**	0.28	8	-0.92	8	**0.65**		773	60	68	**36**	**2.00**	3	-1.30	8	-0.70		53	18	28	**36**	**2.00**	8	-0.17	4	-1.83
		3	250	60	10	7	**1.58**	**8**	-1.54	**8**	-0.04		973	60	54	**36**	**1.90**	25	-0.45	8	-1.45		58	16	28	**36**	**2.00**	8	-0.47	6	-1.53
		4	350	60	6	**16**	**1.66**	8	-1.63	8	-0.03		1373	60	38	**36**	**1.34**	14	0.55	8	-1.89		68	14	28	**36**	**2.00**	8	-0.58	4	-1.42
		5	400	60	5	**14**	**1.70**	8	-1.54	8	-0.16		1573	60	34	**36**	**1.12**	4	0.77	8	-1.89		73	13	28	**36**	**2.00**	8	-0.55	7	-1.45
		6	450	60	4	**14**	**1.34**	8	-1.46	8	0.11		1773	60	30	**36**	**1.94**	10	-0.55	8	-1.39		78	12	28	**36**	**2.00**	8	-0.30	7	-1.70
	No. 2 *P*	7	300	30	8	**22**	**1.12**	8	-1.23	8	0.10	No. 2 *P*	1173	30	45	**36**	**2.00**	11	-1.96	8	-0.04	No. 2 *C* and *itr*	48	15	37	**36**	**1.97**	8	-0.48	7	-1.49
		8	300	40	8	**9**	**1.57**	8	-1.54	8	-0.03		1173	40	45	**36**	**2.00**	9	-1.96	8	-0.04		53	15	33	**36**	**2.00**	8	-0.26	9	-1.74
		9	300	50	8	**25**	**1.03**	8	-1.23	8	0.19		1173	50	45	**36**	**2.00**	5	-1.87	8	-0.13		58	15	30	**36**	**2.00**	8	-0.42	10	-1.58
		10	300	70	8	**14**	**1.89**	8	-1.64	8	-0.25		1173	70	45	**36**	**2.00**	5	-1.78	8	-0.22		68	15	26	**36**	**2.00**	8	-0.57	7	-1.43
		11	300	80	8	4	**1.89**	**8**	-1.64	**8**	-0.25		1173	80	45	**36**	**2.00**	8	-1.98	8	-0.03		73	15	24	**36**	**2.00**	8	-0.41	9	-1.59
		12	300	90	8	**10**	**0.89**	8	-1.14	8	0.26		1173	90	45	**36**	**2.00**	5	-1.75	8	-0.25		78	15	22	**36**	**2.00**	8	-1.47	1	-0.53
	–	13	–									–	–									No. 3 *P* and *itr*	63	5	84	**36**	**2.00**	8	-0.31	15	-1.69
		14																					63	10	42	**36**	**2.00**	8	-0.57	6	-1.43
		15																					63	20	21	**36**	**2.00**	8	-0.56	4	-1.44
		16																					63	25	17	**36**	**2.00**	8	-0.48	8	-1.52
		17																					63	30	14	**36**	**2.00**	8	-0.47	8	-1.54
		18																					63	35	12	**36**	**2.00**	8	-0.73	6	-1.27
NA3	No. 1 *G* and *itr*	1	150	60	19	**17**	**1.78**	15	-1.65	11	-0.13	No. 1 *G* and *itr*	455	60	193	**17**	**1.80**	10	-1.17	11	-0.63	No. 1 *C* and *P*	74	21	28	17	**1.95**	15	-1.03	**19**	-0.92
		2	200	60	13	12	**1.85**	**15**	-1.66	11	-0.19		955	60	92	**17**	**1.72**	5	-0.61	11	-1.11		84	19	28	**17**	**1.75**	15	-1.67	8	-0.09
		3	250	60	10	**16**	**1.56**	15	-1.63	11	0.07		1455	60	61	**17**	**1.71**	8	-1.07	11	-0.65		94	17	28	**17**	**1.84**	15	-1.61	9	-0.23
		4	350	60	6	13	**1.86**	**15**	-1.66	11	-0.20		2455	60	36	**17**	**1.83**	16	-1.73	11	-0.10		114	14	28	**17**	**1.91**	15	-1.49	10	-0.42
		5	400	60	5	**17**	**1.97**	15	-1.67	11	-0.29		2955	60	30	**17**	**1.79**	8	-1.26	11	-0.53		124	13	28	**17**	**1.85**	15	-1.61	12	-0.24
		6	450	60	4	12	**1.35**	**15**	-1.57	11	0.23		3455	60	25	17	**1.79**	**18**	-1.50	11	-0.29		134	12	28	**17**	**1.84**	15	-1.69	15	-0.15
	No. 2 *P*	7	300	30	8	10	**1.66**	**15**	-1.64	11	-0.02	No. 2 *P*	1955	30	45	**17**	**1.82**	15	-1.33	11	-0.50	No. 2 *C* and *itr*	74	15	40	**17**	**1.94**	15	-1.42	14	-0.51
		8	300	40	8	6	**1.88**	**15**	-1.66	11	-0.22		1955	40	45	**17**	**1.81**	8	-1.32	11	-0.49		84	15	35	**17**	**1.95**	15	-0.88	12	-1.07
		9	300	50	8	**15**	**1.86**	**15**	-1.66	11	-0.20		1955	50	45	**17**	**1.57**	7	-0.49	11	-1.08		94	15	31	**17**	**1.86**	15	-1.77	9	-0.09
		10	300	70	8	**22**	**1.75**	15	-1.64	11	-0.11		1955	70	45	**17**	**1.79**	9	-1.24	11	-0.56		114	15	26	17	**1.87**	15	-1.68	**18**	-0.19
		11	300	80	8	12	**1.97**	**15**	-1.67	11	-0.29		1955	80	45	**17**	**1.78**	7	-1.24	11	-0.54		124	15	24	**17**	**1.68**	15	-1.61	10	-0.07
		12	300	90	8	10	**1.72**	**15**	-1.64	11	-0.08		1955	90	45	17	**1.79**	**18**	-1.50	11	-0.29		134	15	22	17	**1.95**	15	-0.48	**30**	-1.47
	–	13	–									–	–									No. 3 *P* and *itr*	104	5	85	**17**	**1.90**	15	-1.41	13	-0.50
		14																					104	10	42	**17**	**1.65**	15	-1.58	9	-0.07
		15																					104	20	21	**17**	**1.47**	15	-1.84	9	0.37
		16																					104	25	17	17	**1.88**	15	-1.39	**20**	-0.49
		17																					104	30	14	17	**1.76**	15	-1.51	**19**	-0.25
		18																					104	35	12	**17**	**1.71**	15	-1.68	7	-0.04

6. Conclusions and Future Research Directions

This paper investigated a multi-objective DNDP, considering reserve capacity and two newly proposed travel time related objective functions. In this discrete network design problem, four types of decisions are made, including adding lanes to the existing network links, constructing new links, determining the lane allocations on two-way links, and converting some two-way links to one-way links. Two variations of the problem are defined based on different restrictions in the number of lanes in each direction of two-way links. The two variations are modeled as mathematical problems with equilibrium constraints, which are bi-level in nature with two lower level problems, one concerning the traffic flows for the future demand and the other concerning the traffic flows for the current demand level. The non-convexity and complexity of the two proposed models led us develop and propose three hybrid multi-objective evolutionary algorithms to solve the problem for good rather than exact solutions. They are the hybrid genetic algorithm (GA), the hybrid evolutionary simulated annealing (ESA), and the hybrid of artificial bee colony (ABC) and simulated annealing (SA) algorithm, which are developed using the adaptive weight approach to capture the multi-objective characteristic of the two problems under study.

A measure based on the set coverage concept was also developed to illustrate the quality of the Pareto-optimal solution set obtained by an algorithm compared to the others. Networks of small, medium and large sizes were used for evaluating the effectiveness of the proposed algorithms. The results show that the proposed multi-objective hybrid GA outperforms other algorithms, in terms of values of the objective functions and the quality of the Pareto-optimal solutions. The multi-objective hybrid of ABC and SA is the second preferred solution procedure, and multi-objective ESA is in the third place.

Finally, we propose some suggestions that were not considered in this paper but can be regarded in future research. For example, the intersection designs were not taken into account in this problem. Their inclusion as decision variables and also presuming the time spent on intersections besides the link travel times, like what was done by Lee and Yang (1994), makes the problem closer to the realistic conditions. The problems considering different combinations of these suggestions together with the decision variables discussed in this paper can form interesting and challenging research topics.

Acknowledgements

We are grateful for the constructive comments of reviewers. The last author would like to acknowledge that his research was supported by a grant from the National Natural Science Foundation of China (71271183) and two grants (201011159026, 201111159056) from the University Research Committee of the University of Hong Kong.

References

Abdulaal M., LeBlanc, L.J. (1979) Continuous Equilibrium Network Design Models, *Transportation Research Part B* 13(1):19-32.

Allsop R.E. (1972) Estimating the Traffic Capacity of a Signalized Road Junction, *Transportation Research* 6:245-255.

Alzaqebah M., Abullah S. (2011) Hybrid Artificial Bee Colony Search Algorithm Based on Disruptive Selection for Examination Timetabling Problems, *Lecture Notes in Computer Science* 6831:31-45.

Aydin M.E., Fogarty T.C. (2004) A Distributed Evolutionary Simulated Annealing Algorithm for Combinatorial Optimization Problems, *Journal of Heuristics* 10(3):269-292.

Basturk B., Karaboga D. (2006) An artificial Bee Colony (ABC) Algorithm for Numeric Function Optimization, *IEEE Swarm Intelligence Symposium*, May 12-14, Indianapolis, USA.

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

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

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

Chen O.J., Ben-Akiva M.E. (1998) Game-Theoretic Formulations of Interaction between Dynamic Traffic Control and Dynamic Traffic Assignment, *Transportation Research Record* 1617:179-188.

Chen A., Kim J., Zhou Z., Chootinan P. (2007) Alpha Reliable Network Design Problem, *Transportation Research Record*, 2029:49-57.

Chen A., Subprasom K., Ji Z. (2006) A Simulation-based Multi-objective Genetic Algorithm (SMOGA) for Build-operate-transfer Network Design Problem, *Optimization and Engineering Journal*, 7(3):225-247.

Chen A., Yang C. (2004) Stochastic Transportation Network Design Problem with Spatial Equity Constraint, *Transportation Research Record*, 1882:97-104.

Chong C.S., Low M.Y.H., Sivakumar A.I., Gay K.L. (2007) Using a Bee Colony Algorithm for Neighborhood Search in Job Shop Scheduling Problems, *Proceedings of the 21st European Conference on Modelling and Simulation*.

Dahl G., Minken H. (2008) Methods Based on Discrete Optimization for Finding Road Network Rehabilitation Strategies, *Computers & Operations Research* 35(7):2193-2208.

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

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

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

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

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

Fisk C.S. (1984) Game Theory and Transportation Systems Modeling, *Transportation Research Part B* 18(4-5):301-313.

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

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

Friesz T.L., Harker P.T. (1985) Properties of the Iterative Optimization-Equilibrium Algorithm, *Civil Engineering Systems* 2:142-154.

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

Gao Z., Wu J., Sun H. (2005) Solution Algorithm for the Bi-level Discrete Network Design Problem, *Transportation Research Part B* 39(6):479-495.

Gen M., Cheng R. (2000) *Genetic Algorithms and Engineering Optimization*, John Wiley and Sons, New York, USA.

Harker P.T., Friesz T.L. (1984) Bounding the Solution of the Continuous Equilibrium Network Design Problem”, *Proceedings of the 9th International Symposium on Transportation and Traffic Theory (VNU Science Press*).

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

Karaboga D. (2005) An Idea Based on Honey Bee Swarm for Numerical Optimization. *Technical report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department.

Karaboga D., Basturk B. (2008) On the Performance of Artificial Bee Colony (ABC) Algorithm”, *Applied Soft Computing* 8(1):687-697.

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

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

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

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

Low M., Chandramohan M., Choo C.S. (2009) Application of Multi-Objective Bee Colony Optimization Algorithm to Automated Red Teaming, In *Proceedings of the 2009 Winter Simulation Conference*, 1798-1808.

Luathep P., Sumalee A., Lam W.H.K, Li Z.C., Lo H.K. (2011) Global Optimization Method for Mixed Transportation Network Design Problem: a Mixed-Integer Linear Programming Approach, *Transportation Research Part B* 45(5):808-827.

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

Meng Q., Yang H., Bell M.G.H. (2001) An Equivalent Continuously Differentiable Model and a Locally Convergent Algorithm for the Continuous Network Design Problem, *Transportation Research Part B* 35(1):83-105.

Meng Q., Yang H. (2002) Benefit Distribution and Equity in Road Network Design, *Transportation Research part B* 36(11):19-35.

Meng Q., Khoo H.L. (2008) Optimizing Contraflow Scheduling Problem: Model and Algorithm, *Journal of Intelligent Transportation Systems* 12(3):126-138.

Meng Q., Ling H, Cheu R.L. (2008) Microscopic Traffic Simulation Model-Based Optimization Approach for the Contraflow Lane Configuration Problem, *Journal of Transportation Engineering-ASCE* 134(1):41-49.

Miandoabchi E., Farahani R.Z. (2011) “Optimizing Reserve Capacity of Urban Road Networks in a Discrete Network Design Problem”, *Advances in Engineering Software*, 42(12), 1041-1050.

Miandoabchi E., Farahani R.Z., Szeto W.Y. (2012) Bi-objective Bimodal Urban Road Network Design Using Hybrid Metaheuristics, *Central European Journal of Operations Research* 20(4):583-621.

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

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

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

Pacheco J., Alvarez A., Casado S., González-Velarde J.L. (2009) A Tabu Search Approach to an Urban Transport Problem in Northern Spain, *Computers & Operations Research* 36(3):967-979.

Pham D., Ghanbarzadeh A. (2007) Multi-Objective Optimisation using the Bees Algorithm, In *Proceedings of the 3rd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2007)*, Whittles, Dunbeath, Scotland.

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

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

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

Russo F., Vitetta A. (2006) A Topological Method to Choose Optimal Solutions after Solving the Multi-

Criteria Urban Road Network Design Problem, *Transportation* 33(4):347-370.

Sadiq A.T., Hamad A.G. (2010) BSA: A Hybrid Bees' Simulated Annealing Algorithm to Solve Optimization & NP-Complete Problems, *Engineering & Technology Journal* 28(2):271-281.

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

Steenbrink P.A. (1974a) *Optimization of Transportation Networks*, John Wiley & Sons, New York, USA.

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

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

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

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

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

Szeto W.Y., Wu Y.Z. (2011) A Simultaneous Bus Route Design and Frequency Setting Problem for Tin Shui Wai, Hong Kong, *European Journal of Operations Research* 209(1):141-155.

Szeto W.Y., Wu Y.Z., and Ho S.C. (2011) An Artificial Bee Colony Algorithm for the Capacitated Vehicle Routing Problem, *European Journal of Operational Research*, 215:126-135.

Szeto W.Y., Jiang Y. (2012) Hybrid Artificial Bee Colony Algorithm for Transit Network Design. *Transportation Research Record,* 2284: 47-56.

Teodorović D., Dell'orco M. (2005) Bee Colony Optimization - A Cooperative Learning Approach to Complex Transportation Problems, *Advanced OR and AI Methods in Transportation*, *Proceedings of 10th EWGT Meeting*, *16th Mini-EURO Conference*.

Teodorović D., Edara P. (2007) A Real-Time Road Pricing System: The Case of a Two-Link Parallel Network, *Computers and Operations Research* 34(1):2-27.

Teklu F., Sumalee A., Watling D.P. (2007) A Genetic Algorithm Approach for Optimising Traffic Control Signals Considering Routing, *Journal of Computer-Aided Civil and Infrastructure Engineering* 22(1):31-43.

Tobin R.L., Friesz T.L. (1988) Sensitivity Analysis for Equilibrium Network Flow, *Transportation Science* 22(4):242-250.

Wardrop J.G. (1952) Some Theoretical Aspects of Road Traffic Research, *Proceedings of the Institute of Civil Engineers Part II*, 1:325-378.

Webster F.V., Cobbe B.M. (1966) Traffic signal, *Road Research Technical Paper* No.*56*, HMSO, London, UK.

Wong S.C. (1996) On the Reserve Capacities of Priority Junctions and Roundabouts, *Transportation Research Part B* 30(6):441-453.

Yang H. (1995) Heuristic Algorithms for the Bilevel Origin-destination Matrix Estimation Problem, *Transportation Research Part B* 29(4):231-242.

Yang H., Bell M.G.H. (1998a) Models and Algorithms for Road Network Design: A Review and Some New Developments, *Transport Reviews* 18(3):257-278.

Yang H., Bell M.G.H. (1998b) A Capacity Paradox in Network Design and How to Avoid it, *Transportation Research Part A* 32(7):539-545.

Yang H., Yagar S., Iida Y., Asakura Y. (1994) An Algorithm for the Inflow Control Problem on Urban Freeway Networks with User-Optimal Flows, *Transportation Research Part B*, 28(2):123-139.

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

Yang X.S. (2005) Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms, *Lecture Notes in Computer Science* 3562:317-323.

Zhang H., Gao Z. (2009) Bilevel Programming Model and Solution Method for Mixed Transportation Network Design Problem, *Journal of Systems Science and Complexity* 22:446-459.

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

Ziyou, G., Yifan, S. (2002) A Reserve Capacity Model of Optimal Signal Control with User-Equilibrium Route Choice, *Transportation Research Part B*, 36(4):313-323.