Time-Dependent Discrete Road Network Design with both Tactical and Strategic Decisions

Abstract

This paper aims to model and investigate the discrete urban road network design problem, using a multi-objective time-dependent decision making approach. Given a base network made up with two-way links, candidate link expansion projects, and candidate link construction projects, the problem determines the optimal combination of one-way and two-way links, the optimal selection of capacity expansion projects, and the optimal lane allocations on two-way links over a dual time scale. The problem considers both the total travel time and the total CO emissions as the two objective function measures. The problem is modeled using a time-dependent approach which considers a planning horizon of multiple years and both morning and evening peaks. Under this approach, the model allows determining the sequence of link construction, the expansion projects over a predetermined planning horizon, the configuration of street orientations, and the lane allocations for morning and evening peaks in each year of the planning horizon. This model is formulated as a mixed-integer programming problem with mathematical equilibrium constraints. In this regards, two multi-objective metaheuristics, including a modified non-dominated sorting genetic algorithm (NSGA-II) and a multi-objective B-cell algorithm, are proposed to solve the above-mentioned problem. Computational results for various test networks are also presented in this paper.

Keywords: Urban road network design; Time-dependent; Dual time scale; Multi-objective; Evolutionary metaheuristics; Vehicle emissions

1. Introduction

The Network Design Problem or NDP in short refers to the decision making problem that involves the determination of the optimal planning and the management decisions for transport networks. The NDP was extensively reviewed by several authors (e.g., Yang and Bell 1998, Farahani *et al*. 2013).

The NDP ideally fits to the class of Stackelberg (or leader-follower) games. In the NDP, the authority is the leader in the game and responsible to plan and manage the road network, while the network users are the followers and they respond to the decision made by the leader. The problem can be expressed mathematically as a bi-level-programming problem, where the upper level problem describes the decision-making problem of the leader who takes into account the user response, and the lower level problem describes the user response to network design scenarios. The problem can be reformulated into a single-level problem by expressing the lower level problem as the constraints.

More than one way can be used to classify the NDP. Magnanti and Wong (1984) proposed that the NDP could be classified into strategic (e.g., street expansions or constructions), tactical (e.g., street orientations and lane allocations), and operational (e.g., signal setting and toll setting) decision levels, in which the strategic, tactical, and operational levels deal with the long-term, mid-term, and short-term issues, respectively. Another way to classify the NDP is based on the decisions made for network topology and network parameters. Under this classification, there are three forms of the problem, namely the Continuous Network Design Problem (CNDP), the Discrete Network Design Problem (DNDP), and the Mixed Network Design Problem (MNDP). The CNDP (e.g., Meng and Yang, 2002) takes the network topology as a given, and determines the network parameters (e.g., signal timing setting, toll setting, and capacity expansion) that only involves continuous decision variables. The DNDP (e.g., Drezner and Wesolowsky, 1997) considers the network topology (e.g., street constructions and street orientations) and only involves discrete design decision variables. The MNDP (e.g., Cantarella and Vitetta, 2006) deals with the network topology and parameterization and involves both discrete and continuous network design variables.

An observation found in the current NDP literature is that the combination of two or more strategic and tactical decisions has been addressed in a number of DNDPs and MNDPs (e.g., Cantarella and Vitetta, 2006; Miandoabchi and Farahani, 2011; Miandoabchi *et al.*, 2012, 2013). These decisions have been modeled in the form of conventional single time period models. However, the nature of strategic decisions is different from tactical ones. The strategic decisions are of long-term nature and are implemented in a period of 10 years or even longer. In fact, the set of decisions such as new street construction and street expansion projects are usually planned to be implemented in an interval of many years rather than a single year in the future. On the other hand, tactical decisions such as street orientations and lane allocations are mid-term decisions which are usually made and implemented for a short period of time (e.g., one year). Therefore, these decisions could be made more than one time within a period of multiple years. Thus, a more desirable approach to model such combination of decisions is to model them in a planning horizon with multiple years or periods (i.e., using the time-dependent approach) which allows considering of both long-term and mid-term decisions in a single problem. Nevertheless, the time-dependent approach has never been adopted for combined strategic and tactical decisions.

The second observation on the current NDP literature is that most of the NDP studies consider a single demand matrix as the input to the problem which is usually based on the peak hour estimation. However, it is necessary to identify the morning and the evening peak demand patterns, in particular for work trips, because most of the work trips are designated in the central business district during the morning (AM) peak period while most of these trips are leaving the central business district during the evening (PM) peak period. Moreover, different demand patterns imply different lane allocations. Thus, the lane allocations may need to be changed for both the morning and evening peak periods. This concept has only been addressed in a few studies (e.g., Cantarella and Vitetta, 2006).

The third observation on the NDP literature is that until nowadays, most of the papers only investigated the single-objective NDPs, and only a few studies have investigated the multi-objective NDPs (e.g., Friesz *et al.*, 1993; Meng and Yang, 2002; Cantarella and Vitetta, 2006). Apparently, the nature of the urban transportation related problems is multi-objective which covers a wide range of evaluation criteria. It is therefore essential to consider multiple objectives in NDPs.

The fourth observation is that the most widely used objective functions in single and multiple objective NDPs are total travel time/cost and very few of multi-objective NDPs with strategic and tactical decisions consider vehicle emissions (e.g., Cantarella and Vitetta, 2006, Yin and Lawphongpanich, 2006). In fact, vehicle emissions are harmful to human health (Szeto *et al.*, 2012). Moreover, minimizing travel time is not equivalent to minimizing vehicle emissions because the emission rate (such as the CO emission rate) changes nonlinearly and non-monotonically with the increase in speed and hence travel time. Other than travel time/cost, vehicle emissions should be minimized.

In the light of the above observations, this study addresses the combination of strategic and tactical decisions within a multi-objective time-dependent framework of a discrete network design problem. Specifically, this paper considers the following decisions:

(1) adding lanes to the existing network streets,

(2) constructing new streets,

(3) alteration of some two-way streets to one-way streets, and

(4) lane allocations in two-way streets.

The first two are the strategic decisions and the last two are tactical decisions in the NDP context. Based on our observations as discussed above, a model with the combination of the four decisions using the time-dependent approach is conducted. In this problem, strategic decisions are sequenced to be implemented within multiple years, where tactical decisions are made and implemented once for each year. In other words, the values of tactical decision variables are recomputed at the beginning of each year and remained fixed within that year. Also, this problem considers both the AM and PM peak period demand matrices, which implies that the two patterns of tactical decision scenarios are designed for each year. This problem is more realistic, but has a higher degree of complexity as compared to the other similar problems in the literature which arises from the inclusion of various dimensions in the problem. Two objectives are considered for the problem; 1) the minimization of the total travel time and 2) the minimization of total CO emissions, which is one of the major emissions indicator commonly used in the literature (Cantarella and Vitetta, 2006; Yin and Lawphongpanich, 2006). The first objective accounts for traveler's mobility and consumer surplus, and the second accounts for the air pollution exposed by the community (including users and non-users).

The problem is modeled as a mathematical program with equilibrium constraints (MPEC) which regarding to its nature is a bi-level Stackelberg game model. Due to the intrinsic complexity and non-convexity of the proposed problem, we propose two multi-objective evolutionary algorithms as the solution methods to obtain the set of Pareto-optimal solutions. They are the improved versions of NSGA-II and B-cell algorithm. In particular, we develop an improved version of NSGA-II algorithm that uses a different density measure and an improved evolution strategy, and develop a novel version of B-cell algorithm that captures the multi-objective nature of the problem. To the best of our knowledge, this B-cell algorithm version is the first multi-objective version.

Table 1 compares the attributes of the proposed problem with the most similar previous studies. ANN stands for Artificial Neural Network. The DNDP-T, CNDP-T, and MNDP-T are, respectively, the time-dependent extensions of the DNDP, the CNDP and the MNDP. According to this table, it can be concluded that the proposed problem is more complicated than the others in the literature because more realistic decisions and objectives are considered.

Table 1. Comparison of the proposed problems in this study with the previous studies

Reference	NDP Type	Modeling Approach		Decisions					Objective Functions			Two Peak hour Demands	Approach to obtain objective function values
		Single Time Period	Time-Dependent	Street Orientation	Lane Allocation	Street Expansion	Street Construction	Other	Total Travel Time	CO Emissions	Other		
This paper	DNDP-T	-	✓	✓	✓	✓	✓	-	✓	✓	-	✓	Directly solving lower level problem
Miandoabchi *et al.* (2013)	DNDP	✓	-	✓	✓	✓	✓	-	-	-	Reserve capacity, two travel time related functions	-	Directly solving lower level problems
Miandoabchi *et al.* (2012)	Bi-modal DNDP	✓	-	✓	✓	✓	✓	Allocating bus lanes	-	-	Total user benefit, Bus demand share	-	Directly solving lower level problems
Miandoabchi and Farahani (2011)	DNDP	✓	-	✓	✓	✓	-	-	-	-	Reserve capacity	-	Directly solving lower level problems
Szeto *et al.* (2010)	DNDP-T		✓				✓	Toll setting			Change in social surplus		Directly solving the whole problem
Lo and Szeto (2009)	CNDP-T		✓			✓		Toll setting			Consumer surplus		Directly solving the whole problem
Szeto and Lo (2008)	CNDP-T		✓			✓		Toll setting			Social Surplus		Directly solving the whole problem
O'Brien and Szeto (2007)	DNDP-T	-	✓	-	-	✓	-	Toll setting	-	-	Consumer surplus	-	Directly solving the whole problem
Cantarella and Vitetta (2006)	DNDP	✓	-	✓	✓	-	-	Signal setting parking space	✓	✓	Bus, pedestrian and people related	✓	Directly solving lower level problems
Lo and Szeto (2004)	CNDP-T	-	✓	-	-	✓	-	-	-	-	Consumer surplus	-	Directly solving the whole problem
Wei and Schonfeld (1993)	DNDP-T	-	✓	-	-	✓	-	-	✓	-	-	-	Approximating by ANN

To summarize, the paper provides contributions to the literature in three aspects:

1. Propose a new time-dependent model for determining both strategic and tactical decisions in road network design, which allows more realistic modeling of the combined decisions with the joint consideration of the different time horizons for each decision type and demand patterns for both morning and evening peaks;
2. Develop an improved version of NSGA-II algorithm which uses a different density

measure and an improved evolution strategy, and

3. Develop a novel version of B-cell algorithm that captures the multi-objective nature of the problem.

The remainder of the paper is organized as follows. The next section provides the problem definitions. Section 3 is dedicated to the notations and mathematical formulation of the proposed problem. Section 4 discusses the approach used to solve the problem. Section 5 depicts the similarities and differences between the proposed algorithms. Sections 6-7 describe the two proposed solution procedures separately. Section 8 contains the computational results and a performance comparison of the algorithms. Finally, conclusions and future research suggestions are made in Section 9.

2. Problem Definition

The problem under consideration is to design an urban road network, by determining the optimal combination of one-way and two-way links, link and lane additions, and lane allocations on two-way links under a multi-objective decision making framework. Link expansions are considered as adding extra lanes to the existing road network. Indeed, this problem is a DNDP as all the decisions involved can be represented by discrete values. Before proceeding to the problem definition, it is necessary to provide a clear definition of the network elements. In this paper, all types of streets and roads in the network are referred to as "link", as the counterpart of the term "edge" in the mathematical definition of the network. Each link consists of two "arcs" if it is two-way and one arc if it is one-way. Each arc on a link is characterized by a set of lanes, where the number of lanes on an arc defines the flow capacity of the arc. If a movement is not allowed in one direction of a link, then no arc will be found exist in that direction.

2.1. Assumptions

The main assumptions for the studied problem are listed below:

- A basic network with all two-way links exists in advance;
- The travel demand matrices for the AM and PM peak periods of each year are known and fixed;
- The sequences of the strategic decisions are determined over multiple year periods and the tactical decisions are determined for each year, one for the AM peak period and the other for the PM peak period;
- A lane addition or street construction project starts and ends at the same year;
- The user flows are assigned to the network according to the user equilibrium principle.

2.2. Inputs

After describing the problem and the related assumptions, we present the following required inputs for the problem:

- Estimated travel demand matrices for each year;
- Attributes of network links such as capacities, investment costs, and travel time functions.
- Candidate link construction projects (a link construction project refers to the construction of a new street between a pair of nodes in the network). The candidate link construction projects consist of a set of possible street constructions that has been defined by the network authority;
- Candidate lane addition projects (a lane addition project refers to the construction of new lanes adjacent to the existing lanes on the streets). The candidate lane addition projects are determined by the network authority;
- Maximum number of possible lanes added to the existing network links, which depends on the availability of vacant land adjacent to the existing streets, and
- Yearly budget available for lane addition and link construction projects.

2.3. Outputs

The following are the outputs obtained from the problem:

- The set of lane addition projects to be executed in each year,
- The set of new street construction projects to be implemented in each year,
- Orientation of one-way links during both AM and PM peak periods for each year, and
- The number of lanes allocated to each direction of each of the two-way links during both the AM and PM peak periods for each year.

Two objective functions are considered for the problem. The first objective function is defined by the conventional total travel time of all travelers. It is the sum of total travel times over the planning horizon. The total travel times for each year is obtained by summing up the AM and PM peak hour travel times of that year. The second objective function measures the total CO emissions over the planning horizon, which is obtained by summing up the CO emissions for each peak period and each planning year.

3. Mathematical model and notations for DNDP-T

The following are the notations 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 corresponding to existing network link l

$S'_{l'}$: set of arcs corresponding to candidate network link l'

W: set of all origin-destination (OD) pairs

U: set of design years, $\tau = 1,\ldots,T$

V: set of daily peak periods (i.e., morning and evening periods)

3.2. Variables

y^τ_l: number of lanes added to existing link l in year τ

$u^\tau_{l'}$: binary variable, which equals 1 if link l' is built in year τ, and zero otherwise

$z_{ij}^{\tau\omega}$: binary variable, which equals 1 if arc (i, j) is built or present (i.e., traffic is allowed in that direction) during daily peak period ω in year τ, and zero otherwise

$k_{ij}^{\tau\omega}$: number of lanes allocated to arc (i, j) during daily peak period ω in year τ

$x_{ij}^{\tau\omega}, x_{ij}^{\tau\omega*}$: traffic flow and user equilibrium traffic flow on arc (i, j) during daily peak period ω in year τ

$X_r^{\tau\omega}$: user equilibrium flow on route $r \in R$ during daily peak period ω in year τ

$\mathbf{\upsilon}^{\tau\omega}$: vector of design variables representing a design scenario during daily peak period ω in year τ, $\mathbf{\upsilon}^{\tau\omega} = \left[y_l^\tau, u_{l'}^\tau, z_{ij}^{\tau\omega}, x_{ij}^{\tau\omega}\right]$

R_τ: cumulative budget that has not yet been spent in year τ and is ready to use for year τ+1

$\delta_{ijr}^{\tau\omega}$: binary variable during daily peak period ω in year τ, which equals 1 if route r uses arc (i, j), and zero otherwise

$\sigma_{oo'ij}^{\tau\omega}$: binary variable, which equals 1 if arc (i, j) is on a route between nodes o and o' during daily peak period ω in year τ, and zero otherwise.

3.3. Parameters

$d_{pq}^{\tau\omega}$: travel demand between OD pair (p, q) during daily peak period ω in year τ

$D^{\tau\omega} = [d_{pq}^{\tau\omega}]$: matrix of travel demands during daily peak period ω in year τ

B: annual budget available for lane addition and link construction projects

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

λ_l: length of arc (i, j)

K_l^τ: current number of lanes on existing link *l* in year τ

$K_{l'}^\tau$: number of lanes on new link *l'* in year τ

M: a large positive number

3.4. Functions

$g_l^\tau(y_l)$: investment cost function for the expansion of existing link *l* in year τ, when y_l lanes are added to both sides of the link

$g'^\tau_{l'}(y_{l'})$: investment cost function for the construction of new link *l'* in year τ

$c_{ij}^{\tau\omega}(k_{ij}^{\tau\omega})$: capacity of arc (i, j) during daily peak period ω in year τ (which equals the product of the number of lanes k_{ij} on the arc and the capacity of a lane)

$t_{ij}^{\tau\omega}(x_{ij}^{\tau\omega}, c_{ij}^{\tau\omega})$: travel time function of arc (i, j) during daily peak period ω in year τ

$e_{ij}^{\tau\omega}(t_{ij}^{\tau\omega})$: vehicular CO emission function of arc (i, j) during daily peak period ω in year τ

Z_1: the first objective function - the total travel time over the planning horizon

Z_2: the second objective function - the total CO emissions over the planning horizon

3.5. Mathematical model

The DNDP-T is a bi-level programming problem with an upper and a lower level problem. The upper level problem is a bi-objective mixed integer mathematical problem, and the lower level problem is the conventional deterministic user equilibrium problem. Mathematically, the DNDP-T can be formulated as follows:

$$\text{Min } Z_1 = \sum_{\tau \in U} \sum_{\omega \in V} \sum_{(i,j) \in A \cup A'} t_{ij}^{\tau\omega} \, x_{ij}^{\tau\omega *} \qquad (1)$$

$$\text{Min } Z_2 = \sum_{\tau \in U} \sum_{\omega \in V} \sum_{(i,j) \in A \cup A'} e_{ij}^{\tau\omega} x_{ij}^{\tau\omega *} \qquad (2)$$

Subject to

$$\sum_{l \in L} g_l^1(y_l) + \sum_{l' \in L'} g'^1_{l'}(y_{l'}) + R_1 = B \qquad (3)$$

$$\sum_{l \in L} g_l^\tau(y_l) + \sum_{l' \in L'} g'^\tau_{l'}(y_{l'}) + R_\tau = R_{\tau-1} + B \qquad \forall \tau > 1 \qquad (4)$$

$$0 \le \sum_{\tau \in U} y_l^\tau \le y_l^{\max} \qquad \forall l \in L \qquad (5)$$

$$k_{ij}^{\tau\omega} + k_{ji}^{\tau\omega} = K_l^\tau + 2y_l^\tau \quad \forall l \in L, \exists! \, (i,j) \in S_l, \tau \in U, \omega \in V \qquad (6)$$

$$k_{ij}^{\tau\omega} + k_{ji}^{\tau\omega} = K_{l'}^\tau . u_{l'}^\tau \qquad \forall l' \in L', \exists! \, (i,j) \in S'_{l'}, \tau \in U, \omega \in V \qquad (7)$$

$$z_{ij}^{\tau\omega} \leq k_{ij}^{\tau\omega} \qquad \forall (i,j) \in A \cup A', \tau \in U, \omega \in V \tag{8}$$

$$k_{ij}^{\tau\omega} \leq M.z_{ij}^{\tau\omega} \qquad \forall (i,j) \in A \cup A', \tau \in U, \omega \in V \tag{9}$$

$$z_{ij}^{\tau\omega} + z_{ji}^{\tau\omega} \geq 1 \qquad \forall (i,j) \in A, \tau \in U, \omega \in V \tag{10}$$

$$z_{ij}^{\tau\omega} + z_{ji}^{\tau\omega} \geq u_{l'}^{\tau} \qquad \forall (i,j) \in A', l' \in L', \tau \in U, \omega \in V \tag{11}$$

$$z_{ij}^{\tau\omega} + z_{ji}^{\tau\omega} \leq M.u_{l'}^{\tau} \qquad \forall (i,j) \in A', l' \in L', \tau \in U, \omega \in V \tag{12}$$

$$K_l^{\tau+1} = K_l^{\tau} + 2y_l^{\tau} \qquad \forall l \in L, \tau \in U, \tau > 1 \tag{13}$$

$$u_{l'}^{\tau+1} \geq u_{l'}^{\tau} \qquad \forall l' \in L', \tau \in U, \tau > 1 \tag{14}$$

$$\sum_{(p,j) \in A \cup A'} \sigma_{pqpj}^{\tau\omega} = 1 \quad \forall (p,q) \in W, \tau \in U, \omega \in V \tag{15}$$

$$\sum_{(i,q) \in A \cup A'} \sigma_{pqiq}^{\tau\omega} = 1 \quad \forall (p,q) \in W, \tau \in U, \omega \in V \tag{16}$$

$$\sum_{(i,j) \in A \cup A', i \neq j} \sigma_{pqij}^{\tau\omega} = \sum_{(j,q) \in A \cup A', j \neq q} \sigma_{pqjq}^{\tau\omega} \quad \forall j \in N, (p,q) \in W, \tau \in U, \omega \in V \tag{17}$$

$$\sigma_{pqij}^{\tau\omega} \leq M.z_{ij}^{\tau\omega} \quad \forall (i,j) \in A \cup A', (p,q) \in W, \tau \in U, \omega \in V \tag{18}$$

$$k_{ij}^{\tau\omega}, y_l^{\tau} \geq 0, \text{ are integers} \quad \forall (i,j) \in A \cup A', l \in L, \tau \in U, \omega \in V \tag{19}$$

$$z_{ij}^{\tau\omega}, u_{l'}^{\tau}, \sigma_{pqij}^{\tau\omega} \in \{0,1\} \quad \forall (i,j) \in A \cup A', (p,q) \in W, l' \in L', \tau \in U, \omega \in V \tag{20}$$

$$\sum_{(i,j) \in A \cup A'} t_{ij}^{\tau\omega}\left(x_{ij}^{\tau\omega *}, c_{ij}^{\tau\omega}\right).\left(x_{ij}^{\tau\omega} - x_{ij}^{\tau\omega *}\right) \geq 0 \quad \forall x_{ij}^{\tau\omega} \in \Omega, \tau \in U, \omega \in V \tag{21}$$

$$\Omega = \left\{ x_{ij}^{\tau\omega} \left| \begin{array}{c} \sum_{r \in R_{pq}^{\tau\omega}} X_r^{\tau\omega} = d_{pq}^{\tau\omega} \quad \forall (p,q) \in W; \tau \in U; \omega \in V; \\ x_{ij}^{\tau\omega} = \sum_{(p,q) \in W} \sum_{r \in R_{pq}^{\tau\omega}} X_r^{\tau\omega}.\delta_{ijr}^{\tau\omega} \quad \forall (i,j) \in A \cup A'; \tau \in U; \omega \in V; \\ X_r^{\tau\omega} \geq 0 \end{array} \right. \right\} \tag{22}$$

The model considers two objectives: the first objective is to minimize the total travel time and the second objective is to minimize total CO emissions. Constraints (3)-(4) impose a yearly budget limit on the sum of construction costs during each design year. It is assumed that the unspent budget of each year will be available to be spent in the next years. Thus, except for the base year in which only yearly budget is available, for other years, the cumulative budget left after a year is added to the yearly budget of the following year. Constraint (5) implies that the total number of lanes to be added on each side of an existing link during the design years is restricted by a maximum number of lanes allowed to be built. Constraints (6)-(7) allocate the total number of lanes on each existing link and new link for two daily peak periods in each year, in which the numbers of lanes added to both directions are the same. Note that ∃! means 'there exists exactly one' and in these constraints, for any *l* or *l'*, there is exactly one pair of nodes (*i*, *j*) corresponding to *l* or *l'*. Constraints (8)-(9) avoid allocating lanes to the arcs that are not present in the network. Constraint (10) ensures that traffic is not blocked on both directions of the existing links. Constraints (11)-(12) assure that traffic is not blocked on both directions of the new links. Constraint (13) implies the number of links on an existing link in year τ+1 is equal to that in year τ plus the number of lanes added in year τ. Constraint (14) ensures that if a new link is added in year τ, it remains available in the following years.

The set of constraints (15)-(18) ensures the connectivity of the design scenario, by ensuring the existence of a route between each OD pair in the network for each peak hour period and each year. Constraints (15) and (16) ensure that the first and the last arcs of the route start from and ends at the origin and destination nodes, respectively. Constraint (17) defines the inner parts of the route, by finding a sequence of connected arcs between the first and last arcs. Constraint (18) ensures that no route can pass a non-existing arc. For a feasible design scenario, constraints (15)-(18) must be satisfied for all OD pairs, both peak hour periods, and all years.

Constraints (19)-(20) define the variable domains. Constraints (21)-(22) are the variational inequalities formulations for the user equilibrium traffic assignment problems.

The functional form of $e_{ij}^{\tau\omega}$ in (2) is adopted from the literature (e.g., Yin and Lawphongpanich, 2006):

$$e_{ij}^{\tau\omega}(t_{ij}^{\tau\omega}) = 0.2038.\, t_{ij}^{\tau\omega}.\, e^{0.7962.(\lambda_{ij}/t_{ij}^{\tau\omega})} \tag{23}$$

Equation (23) is equivalent to the function used in TRANSYT-7F (e.g., Rilett and Benedek, 1994), but the coefficients are computed in kilometer and minutes instead of feet and seconds. Vehicular CO emissions associated with a particular arc (i, j) with the length λ_{ij} is in grams per vehicle and the total CO emissions is in grams per hour.

4. The Solution Approach and the Proposed Algorithms

We develop efficient solution methods to find a good and nearly global optimal solution rather than an exact and global optimal solution for the present DNDP-T mathematical model due to the following reasons:

- Bi-level programming problems in general are NP-hard. A study conducted by Ben-Ayed *et al.* (1988) showed that even a simple linear bi-level programming problem is still NP-hard;
- Many bi-level programming problems are non-convex. Even if both of the upper and lower level problems are convex, it is not guaranteed that the whole problem is convex;
- Though the Branch and Bound algorithms or the enumerative algorithms can be applied to get global optimal solutions to the two problems with small size networks, they are not able to solve real size network problems, and
- There has been an increasing interest recently in solving NDPs by metaheuristics.

In particular, two multi-objective population-based metaheuristics are proposed to solve the problem: an improved version of Non-dominated Sorting Genetic Algorithm (NSGA-II) and a new multi-objective version B-cell Algorithm (mBCA). To the best of our knowledge, the B-cell algorithm has never been applied to solve the urban transportation network design problems. The proposed versions of both algorithms are novel. The proposed NSGA-II employs a new solution density measure, and the mBCA is novel in the sense that it is the first multi-objective version for this algorithm and it consists of novel features compared with the original one. In the following sections, the similarities and comparisons of the two algorithms are presented in Section 5 and the illustrations on the two proposed algorithms are detailed in Sections 6-8.

5. Comparisons of the Proposed Algorithms

The two algorithms are similar in the view of checking for the strong connectivity of networks, obtaining the objective function values, initial population generation, and Pareto-optimal solution set generation.

5.1. Checking the Feasibility of Solutions

A solution is considered feasible if it satisfies two criteria. The first criterion is the budget feasibility of the solution which means that its expansion and construction project execution plan satisfies constraints (3)-(4). The second criterion is the strong connectivity of OD pairs in all of its design scenarios for each peak period and each design year. A solution is considered to be infeasible in terms of connectivity if at least one OD pair of at least one of the $2T$ networks is disconnected. The strong connectivity of each network is checked in two stages:

Stage 1:

- Perform test 1: Check if all nodes have at least one outgoing and one incoming lanes (necessary condition of network connectivity).

Stage 2:

- If the solution passes test 1, perform test 2: check if there is a (shortest) path between each OD pair using Dijkstra's algorithm;
- If the solution passes test 2, accept the solution; otherwise reject the solution.

Indeed, if at least one of the OD pairs is disconnected, i.e., no shortest path can be found between them, then the created network is rejected because of its disconnectedness. The two phase connectivity checking helps to avoid the unnecessary use of the shortest path algorithm which is time consuming for medium and large sized networks.

If the solution is proved to be feasible, the algorithms will proceed to solve the lower level problem to obtain the user equilibrium flows and then the objective function values.

5.2. Obtaining the Objective Function Values

Calculating the objective function values requires solving the deterministic user equilibrium lower level problems. Many studies on NDPs are single time period problems which require solving one to two lower level problems. However, for time-dependent NDPs such as the problem under this study, the number of lower level problems tends to be very large depending on the specific dimensions of the problem.

For the problem in this paper, there are two time dimensions, namely daily peak and year. For each solution, it is required that the deterministic user equilibrium problem is solved for each design scenario in each year and for each peak period. This implies that $2T$ lower level problems are solved for each solution. This can be done by separately solving the user equilibrium traffic assignment problem (21)-(22) for a specific design scenario. The traffic assignment problem (21)-(22) is in fact a nonlinear convex problem which can be solved by many different methods, such as a common convex-combination-based algorithm called the Frank-Wolfe (FW) method (see Sheffi (1985) for the details). It should be noted that although the path flow variables are used in the lower level problem (21)-(22), the FW method is a link-based algorithm which is based on finding the shortest path between OD pairs. The use of the path flow variables in the lower level problem is adopted in this paper, and is for demonstration and modeling purposes only.

5.3. Initial Population Generation

Initial population members are constructed randomly based on a heuristic procedure. The procedure consists of two phases: 1) In the first phase, a feasible plan of expansion and construction projects is generated. 2) In the second phase, with the total number of lanes on each link, a random but feasible lane allocation scenario is built for both AM and PM peak periods and each design year τ. The initial solution generation procedure is described below:

Phase 1 (Project planning):

- Select the possible expansion and construction projects randomly, until the construction cost reaches the defined budget for the whole planning horizon, i.e., $B \cdot T$.
- Execute a random plan for the implementation of the projects:
  - Define all possible combinations of implementation years for the selected set of projects.
  - Select a set of combinations randomly (e.g., 100 combinations) and check their

budget feasibility according to constraints (3)-(4).

  - Select a random implementation plan from the feasible options.
  - Construct new links and expand existing links in particular year according the selected plan.

Phase 2 (Lane allocation):

- From the available network lanes, allocate the minimum number of lanes to ensure the presence of at least one outgoing and one incoming lanes for all nodes:
    - For each node, add a lane with a random direction on one of its connected links.
    - For nodes with zero incoming lanes, add an incoming lane on one of its connected links.
    - For nodes with zero outgoing lanes, add an outgoing lane on one of its connected links.
- Randomly allocate the remaining lanes to each side of their corresponding link.

The lane allocation procedure can help avoid generating a large number of infeasible solutions and save computational time. Any generated solution which has at least one disconnected network is discarded. Then, the procedure is repeated until the desired number of solutions is generated.

5.4. Pareto-Optimal Solution Set Generation

Because the two developed algorithms are multi-objective, it is required to maintain a set of Pareto-optimal solutions during their solution procedure. Each algorithm handles the Pareto-optimal solutions in different ways. For example, NSGA-II maintains the Pareto-optimal solutions inside the population using a specific mechanism, so that the Pareto-optimal set can be extracted from the population at the end of the solution process. In the mBCA, a separate list of Pareto-optimal solutions is generated and updated throughout the solution procedure. After being checked for non-dominance by other solutions, each of the solutions is added to the Pareto-optimal set, and the dominated solutions are omitted from the Pareto-optimal set if necessary.

5.5. Comparison of the Proposed Algorithms

In order to provide an overall view to the developed algorithms and illustrate their differences, Table 2 shows the comparison of the developed algorithms and their general structural characteristics. The details will be described in later sections.

Table 2. Comparison of the developed algorithms

Algorithm	**Modified NSGA-II**	**mBCA**
Number of iterations	*G* generations	*C* cycles
Solution generation method	Select two parents and apply mutation and crossover	Generate clones for each solution and apply hypermutation
Evolution strategy	Use a sorting mechanism to select solutions to form a new population from the combination of offspring and the current population	Replace the solution with one of its selected mutated clones

6. Modified Non-dominated Sorting Genetic Algorithm II

Genetic algorithm was first introduced by Holland (1975). It is a population-based nature-inspired metaheuristic which simulates the process of genetic evolution. Among the existing metaheuristics, GA and its hybrid extensions have been widely used and have successfully solved NDPs (e.g., Cantarella *et al.*, 2006; Cantarella and Vitetta, 2006; Szeto and Wu, 2011). The well-known multi-objective genetic algorithm (NSGA-II) was introduced by Deb *et al*. (2002). This algorithm has not been used to solve NDPs until recently and is served as a benchmark algorithm. The whole procedure of the algorithm is presented below and the details will be described in later sections.

Phase 1: Generate a population of *P* solutions and rank them. Then, set their fitness values to be equal to their ranks.

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

- Select a pair of parent solutions using the binary tournament selection and the parent selection operator.
- Merge the selected parents by applying a merging process to produce an offspring set.
- Apply the mutation operator on the offspring set.
- Check the connectivity of each offspring solution and discard the solution if it is infeasible.
- Check the budget feasibility of each offspring and apply the budget reduction sub-routine if it is needed.
- Calculate the objective function values for the feasible offspring solutions.
- Form a combined population using the current population and the offspring set.
- Select *P* solutions from the combined population by applying the evolution mechanism.
- Set the *P* solutions as the new population, and then assign their ranks as their fitness values.

Phase 3: Return the set of solutions with rank 1 from the latest population as the Pareto-optimal set.

6.1. Solution Encoding

Chromosome representation is designed in a way such that it encodes the complete configuration of the network over the planning horizon with T design years. Each chromosome is represented by a 4-row matrix as shown in Table 3, in which each column corresponds to one network link for a particular design year and each row corresponds to the lane allocation of a particular arc belonging to a particular link during a particular peak period and a particular design year. $|L|+|L'|$ is the total number of existing and candidate links for each design year in the planning horizon. $k_{ij}^{\tau 1}$ and $k_{ji}^{\tau 1}$ represent the AM peak lane allocations of arcs (i, j) and (j, i) in year τ, respectively while $k_{ij}^{\tau 2}$ and $k_{ji}^{\tau 2}$ represent the PM peak lane allocations of these arcs in the same year. The total length of each chromosome equals $4T(|L|+|L'|)$, and each solution consists of $2T$ network configurations as there are two peak periods for each of the T years.

Table 3. The chromosome representation

	year 1				year 2				…				year T			
	1	2	…	$\|L\|+\|L'\|$	1	2	…	$\|L\|+\|L'\|$	1	2	…	$\|L\|+\|L'\|$	1	2	…	$\|L\|+\|L'\|$
$k_{ij}^{\tau 1}$	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$k_{ji}^{\tau 1}$	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$k_{ij}^{\tau 2}$	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$k_{ji}^{\tau 2}$	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Figure 1 illustrates a typical 7-node and 12-link network and its chromsome representation for the AM and PM peak periods in year τ is presented in Table 4. It is assumed that the network has two candidiate new links, namely links 6-2 and 4-7, which are indicated by thick lines in Figure 1 and in gray columns in Table 4. The dotted and dashed line for link 4-7 indicates that it is NOT constructed in the current network configuration, while the solid line for link 6-2 indicates that it is constructed in the current network configuration. The one-head arrows are one-way links and the two-head arrows are the pairs of one-way arcs. For illustration purposes, the PM network design is obtained by reversing links 1-2 and 3-7 and converting link 6-2 into a two-way link. The complete chromosome repersentation of the network can be obtained by combining all T design scenario chromosomes.

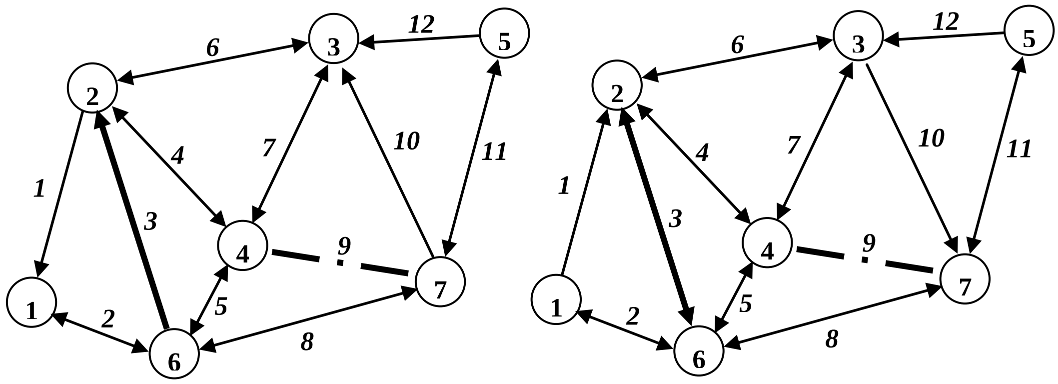

Figure 1. Typical design for AM (left) and PM (right) peak hours in year τ

Table 4. The chromosome representation of design scenarios in Fig. 1

	year τ											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
$k_{ij}^{\tau 1}$	0	1	0	1	1	1	1	1	0	0	1	0
$k_{ji}^{\tau 1}$	2	1	2	1	1	1	1	1	0	2	1	2
$k_{ij}^{\tau 2}$	2	1	1	1	1	1	1	1	0	2	1	0
$k_{ji}^{\tau 2}$	0	1	1	1	1	1	1	1	0	0	1	2

6.2. Calculation of Fitness Value

In this algorithm, the fitness value of a solution is equal to its rank among all the solutions in the population. The ranks of the solutions are determined by sorting the population using the fast non-dominated sorting approach. The sorting mechanism divides the population into non-domination levels. The first non-domination level consists of the solutions which are not dominated by any other solutions in the population and these are called Pareto-optimal solutions. The second non-domination level includes the solutions which are not dominated by any other solutions except by their higher non-domination level (i.e., first level). The rest of the non-domination levels can be defined in the same way. The last non-domination level includes the solutions which are dominated by their higher non-domination level and do not dominate any solutions. In this way, each solution can be assigned a fitness value that equals the non-domination level it belongs to. For instance, the first non-domination level solutions take the rank (or fitness value) 1; the second non-domination level takes the rank 2, etc. The maximum fitness value is equal to the total number of non-domination levels. The detailed explanation of the fast non-dominated sorting can be referred to Deb *et al.* (2002).

6.3. Parent Selection Operator

In the NSGA-II, the parent selection is performed using the binary tournament selection operator. According to this operator, a random pair of solutions is selected and one of them is chosen as the parent. In the original version of the algorithm, the operator uses two types of information for each solution to make the comparison, namely the fitness value (rank) and the crowding-distance. When comparing two solutions obtained from the same non-domination level, the one from a less crowded region is preferred to be chosen. For this purpose, the crowding-distance measure is used to estimate the density of the solutions surrounding a specific solution, by computing the average distance of its two nearest neighbor solutions on the same front located before and after it. For a concerned solution, the value of this measure is computed by summing up the side lengths of the cuboid formed by its neighbor solutions. For the boundary solutions (solutions with the largest and smallest objective function values) of a front, the crowding-distance value is defined as infinity. For a solution $\boldsymbol{\gamma}_p$, the crowding distance measure is calculated by formula (24):

$$Crowding\left(\boldsymbol{\gamma}_p\right) = \sum_{m=1,..,M} \frac{\left|Z_m\left(\boldsymbol{\gamma}_{p_b}\right) - Z_m\left(\boldsymbol{\gamma}_{p_a}\right)\right|}{Z_m^{max} - Z_m^{min}} \tag{24}$$

where Z_m^{max} and Z_m^{min} are the maximum and minimum values of the *m*th objective function in a specific front, respectively, and $\boldsymbol{\gamma}_{p_b}$ and $\boldsymbol{\gamma}_{p_a}$ are the neighbor solutions before and after $\boldsymbol{\gamma}_p$ respectively.

Regarding this measure, a solution with a higher crowding distance value is considered to be in a less crowded region, and is more preferable to the others. However, a solution with a higher crowding distance value does not necessarily mean that the solution is in a low density region because formula (24) does not consider the distance of $\boldsymbol{\gamma}_p$ itself from its neighbor solutions. In other words, although $\boldsymbol{\gamma}_p$ may have a large $Crowding(\boldsymbol{\gamma}_p)$ value, it may be too close to one of its neighbor solutions which means it may be located near a high density region.

Suppose two solutions, $\boldsymbol{\gamma}_p$ and $\boldsymbol{\gamma}_{p'}$, from the same front are compared with each other. Figure 2 illustrates them in a bi-objective solution space. It is observed that $\boldsymbol{\gamma}_p$ has a larger crowding distance value but it is too close to one of its neighbor solutions. In contrast, although $\boldsymbol{\gamma}_{p'}$ has a smaller crowding distance value, it is well located between its neighbor solutions, being not close to any of them. It is evident that $\boldsymbol{\gamma}_{p'}$ is more desirable in terms of density.

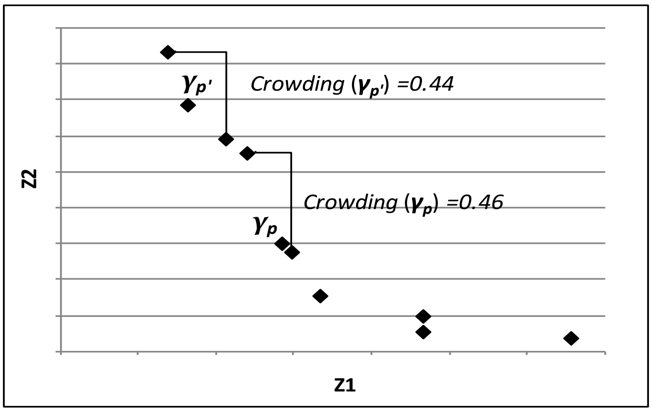

Figure 2. Comparison of two typical solutions for crowding-distance

Given the drawback of the crowding distance measure, in this paper, a new and simple density measure is used to address this issue. The density measure *Den* computes the minimum distance between a solution $\boldsymbol{\gamma}_p$ and other solutions in the same front. The mathematical expression is given below:

$$Den(\boldsymbol{\gamma}_p) = \min_q(Dist(\boldsymbol{\gamma}_p, \boldsymbol{\gamma}_q)) \tag{25}$$

where *Dist* is calculated using the following normalized Manhattan distance equation:

$$Dist(\boldsymbol{\gamma}_p, \boldsymbol{\gamma}_q) = \sum_{m=1,..,M} \frac{\left|Z_m(\boldsymbol{\gamma}_p) - Z_m(\boldsymbol{\gamma}_q)\right|}{Z_m^{max} - Z_m^{min}} \tag{26}$$

The selection rules are as follows:

- If the two solutions γ_p and γ_q belong to different non-domination levels: if $Rank(\gamma_p) < Rank(\gamma_q)$, γ_p is chosen; otherwise, γ_q is selected.
- If γ_p and γ_q belong to the same non-domination level: if $Den(\gamma_p) > Den(\gamma_q)$, γ_p is chosen; otherwise, γ_q is selected.

6.4. Crossover and Mutation

The crossover operator adopted in this paper is a modified form of the operator and was introduced by Drezner and Wesolowsky (1997, 2003), which is a successful merging process that exploits the structure of the problem. The operator attempts to merge their parents in a way that the set of links are taken from each parent and forms a connected set. Such a connected set is built using a "pivot" node. For a pivot node, a link count is calculated for each link of the network, including the set of new links. Then, the link counts are used to construct the offspring. Each node is selected once as the pivot node. In this way, a partitioning scheme is defined for each pivot node. In this paper, the partitioning scheme obtained for each of the *N* nodes as the pivot node is applied to all 2*T* networks to generate 2*N* solutions as the offspring set. Using a unique partitioning scheme for all 2*T* networks is particularly necessary because it allows the uniform selection of expansion and construction projects from each parent. The following procedure is repeated for each pivot node:

- Assign a count *Cnt* = 1 to the links that are directly connected to the pivot node
- Repeat until all links are assigned a count:
  - Select a set of links connected to the links with the count *Cnt*.
  - For each selected link, find the minimum count value ($Cnt_{\min}$) among its connected links, and assign it a count $Cnt = Cnt_{\min}+1$.

- Compute the median of all link counts, Cnt_{med}.
- For each peak period of each design year:
  - Construct the corresponding network of the first offspring solution:
    - Select the design of links with $Cnt < Cnt_{\text{med}}$ from parent 1.
    - Select the design of links with $Cnt > Cnt_{\text{med}}$ from parent 2.
    - Select the design of links with $Cnt = Cnt_{\text{med}}$ randomly from one parent.
  - Construct the corresponding network of the second offspring solution by reversing the direction of one-way links with $Cnt > Cnt_{\text{med}}$ in the first offspring.

To decide whether a new link is included or not, the procedure checks whether the new link is present in the selected parents or not. It is noted that in the original version, a link with $Cnt = Cnt_{\text{med}}$ can be taken independently from either of the parents, but in this paper the whole set of such links are randomly taken from only one parent. The rationale for using this rule is to minimize the possibility of having disconnected solutions. Figure 3 demonstrates an example of link count assignment in a typical network. The pivot node here is 3.

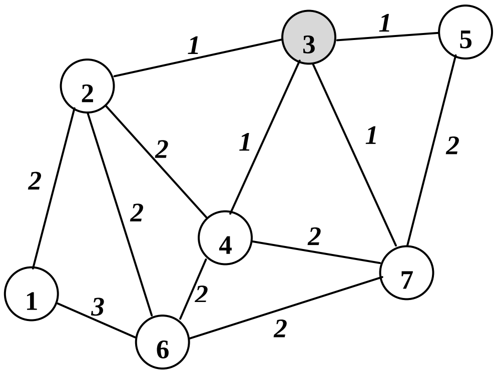

Figure 3. A typical network and its link count assignment

Each generated offspring is exposed to mutation with a specific probability which is a parameter for the algorithm. The mutation operator randomly selects a design year in the solution, and changes the lane allocations of four links (two links in the AM peak network and two in the PM peak network) in that design year randomly.

When changing the allocation of lanes on a link between nodes i and j, a feasibility interval is used to reduce the possibility of generating disconnected networks. The network can become disconnected in two ways: 1) the presence of zero outgoing or incoming lanes for node i or j, and 2) the non-existence of a path between some pairs of nodes even if the first case does not occur. The former can be handled by using a feasibility interval, whereas the latter is not easy to predict and therefore it is not considered in this paper. Because the lane allocation on a link is defined by the number of lanes on arcs (i, j) and (j, i), it suffices to define the interval $[LB_{ij}', UB_{ij}']$ for arc (i, j). By selecting a feasible number of lanes on arc (i, j), the remaining lanes can be allocated to arc

(j, i). In order to compute the interval, the following relations are used:

$$k_{ij} + \sum_{w\in\Phi_i, w\neq j} k'_{iw} \geq 1 \tag{27}$$

$$k_{ji} + \sum_{w\in\Phi_i, w\neq j} k'_{wi} \geq 1 \tag{28}$$

$$k_{ij} + \sum_{w\in\Phi_j, w\neq i} k'_{wj} \geq 1 \tag{29}$$

$$k_{ji} + \sum_{w\in\Phi_j, w\neq i} k'_{jw} \geq 1 \tag{30}$$

$$k_{ij} + k_{ji} = k'_{ij} + k'_{ji} \tag{31}$$

where k'_{iw}, k'_{wi}, k'_{wj}, k'_{jw}, k'_{ij}, and k'_{ji} are the current values of the related lanes allocations, while k_{ij} and k_{ji} are the lane allocation variables for arcs (i, j) and (j, i), respectively. Φ_i and Φ_j are correspondingly the sets of adjacent nodes to nodes i and j. The second terms in (27) and (28) equal the total numbers of outgoing lanes from nodes i and j except the lanes on arcs (i, j) and (j, i), respectively. Similarly, the second terms in (29) and (30) equal the total numbers of incoming lanes to nodes i and j except the lanes on arcs (i, j) or (j, i), respectively. Inequality pair (27)-(28) ensures that at least one outgoing and one incoming lanes remain for node i whereas inequality pair (29)-(30) ensures that at least one outgoing and one incoming lane remains for node j. Equation (31) is the lane allocation constraint. The upper and lower bounds of the interval for k_{ij} are calculated by (32) and (33), which are deduced from conditions (27) and (29) and conditions (28), (30), and (31), respectively. A random value for k_{ij} is chosen from the interval, and then the value of k_{ji} is obtained from the total number of lanes on the link.

$$LB'_{ij} = \max\left\{0, \max\left\{1 - \sum_{w\in\Phi_j, w\neq i} k'_{wj}, 1 - \sum_{w\in\Phi_i, w\neq j} k'_{iw}\right\}\right\} \tag{32}$$

$$UB'_{ij} = \min\left\{k'_{ij} + k'_{ji}, \min\left\{\sum_{w\in\Phi_j} k'_{jw} + k'_{ij} - 1, \sum_{w\in\Phi_i} k'_{wi} + k'_{ij} - 1\right\}\right\} \tag{33}$$

After applying the mutation operator, all offspring solutions are examined for their feasibility; if the offspring is infeasible in terms of the network connectivity of any of the 2T networks, it is discarded. After this process, the offspring is checked for the total construction cost. Once the budget constraints (3)-(4) are violated, a budget reduction sub-routine is applied to repair the infeasibility. The sub-routine is shown below:

- Check whether the current projects can be omitted from the solution.

- If there is at least one feasible omission in terms of network connectivity, repeat the following steps until the budget constraint is not violated or the network becomes disconnected:
    - Omit the project with the maximum cost.
    - If the network becomes disconnected, stop and report as infeasibility. If not, repeat the whole procedure.

6.5. Evolution Mechanism

The evolution process of NSGA-II adopts the non-dominated sorting and the crowded-comparison operator to sort the combined set that is formed by the old population and the offspring set, and to obtain a set of P solutions for the new population. The variation of NSGA-II proposed in this paper is a procedure that aims to select a diverse set of solutions from the last front (if required) using the defined density measure instead of computing the crowding-distance values. The procedure of this evolution mechanism is described as follows:

- Sort all the combined solution sets using non-dominated sorting, and set $P' = P$ as the remaining spaces in the new population.
- Repeat the process until P' becomes zero (i.e., P solutions are inserted into the new population):
    - Select the next best non-dominated front F_i
    - If $|F_i| \leq P'$, add all the solution in F_i to the population and set $P' = P' - |F_i|$; otherwise, select P' solutions from F_i.

The procedure continues to add the solutions belonging to the best remaining non-dominated fronts until the new population is filled with P solutions. If the last front to be selected has less members than the remaining spaces, the required P' solutions are selected from it to include in the population. In the original NSGA-II, such selection is performed by choosing members with maximum crowding-distance values.

In this paper, a new procedure is devised based on the proposed density measure. It guides the algorithm such that a required number of solutions with the largest separation among them is chosen from the last front with a reasonable computational effort. The devised method repeatedly omits the solutions in the front until P' solutions are remained. The basic idea of this method is to omit the solutions that are close to the other solutions so that the remaining P' solutions can be as far as possible to each other at the end. The details of the procedure are given below:

- Calculate the distance between each pair of front solutions using equation (26):
- Repeat until P' solutions are remained in the front:

- Select the solutions pair with the minimum *Dist* value.
- From the pair, select the solution with the minimum *Dist* value to solutions not in the pair, and omit it.
- Update the *Dist* values for the remaining solutions.

Figure 4 illustrates the procedure for the first iteration of the devised method for a typical front with 10 solutions. Experiments with 200-300 random samples for each test problem indicate that the new method leads to a 110% better *Den* value and even a 6.5% better minimum crowding-distance value on average compared to the original method.

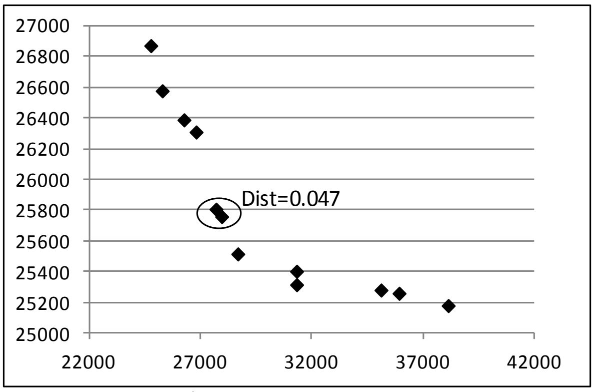

Figure 4(a). 1st iteration: select the closest pair

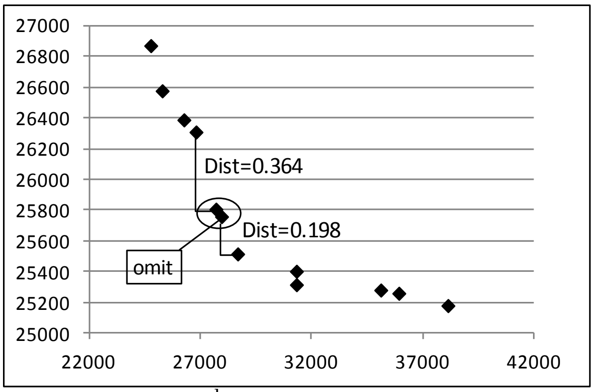

Figure 4(b). 2nd iteration: select for omission

7. Multi-objective B-Cell Algorithm

The B-Cell algorithm which was first proposed by Kelsey and Timmis (2003) and belongs to the group of artificial immune systems. This group encloses a family of algorithms which mimic the behavior of natural immune systems in defending disease-causing organisms. Like all the algorithms of the same category, the B-Cell algorithm procedure relies on the clonal selection principle. This principle is derived from the clonal selection theory which explains how an immune system is stimulated when it is exposed the undesired organisms, how it defeats them, and how to improve its capability by learning from previous fights. According to this principle, when the immune system encounters a disease agent which is generally known as "antigen"', the system at first selects the most appropriate lymphocytes (B-cells) to defeat the agent. In this selection procedure, the system selects the B-cells with their receptors that match the antigens most (i.e., with the highest affinity value). The next procedure is to clone (replicate) the selected B-cells to form an army to fight the invaders. Finally, the B-cell clones undergo contiguous somatic hypermutation and are targeted to enhance their response to the newer antigens. The immune system enhances its defeating capability by repeating the described procedure.

The B-cell algorithm deploys the aforementioned principle, but with some differences in the details. The algorithm maintains a population consisting of a set of solutions (called B-cells) and the solution procedure is repeated for a number of cycles as if other population-based metaheuristics. Each cycle of the algorithm comprises of P iterations, each corresponding to one of the population members. In each iteration, one of the population members is chosen and replicated into a specified number to form the set of so called "cloned solutions". Afterwards, a random solution is inserted in the obtained set. Then, all the cloned solutions plus the inserted solution undergo hypermutation one-by-one to create the set of "mutated clone solutions". The population update is implemented by deciding to replace the concerned population member with the best mutated clone solution, or to maintain it.

The distinct feature of this algorithm is its specific mutation operator which is called "contiguous somatic hypermutation". As stated in Kelsey and Timmis (2003), the algorithm is based on the idea that the cell mutations occur in a cluster of regions in the real immune systems.

This paper develops a multi-objective version of the B-cell algorithm (mBCA) to solve the proposed bi-objective problem. The multi-objective nature of the problem is considered in two features of the algorithm: 1) The algorithm maintains an archive for Pareto-optimal solutions, and 2) the decision to replace each population member with one of its mutated clone solutions is performed by using a set of rules which use the dominance concept and the cooling scheme of the simulated annealing algorithm. The overall procedure of the algorithm is shown below:

Phase 1: Generate a population with P solutions; then, compute the objective function values for the population. Initialize the temperature.

Phase 2: Repeat the following procedure for C cycles:

- For each solution $\boldsymbol{\gamma}_p$, repeat the following procedure:
  - Build the cloned solution set of $\boldsymbol{\gamma}_p$ by replicating it into a number of solutions equal to the value calculated by the affinity based cloning method.
  - Add a random solution to the generated cloned solution set.
  - Apply contiguous somatic hypermutation on each member of the obtained set.
  - Calculate the objective function values for the obtained mutated cloned solutions.
  - Find the Pareto-optimal solutions among the mutated clone solutions and update the Pareto-optimal solutions set.
  - Decide to replace $\boldsymbol{\gamma}_p$ with one of its mutated clone solutions according to the evolution mechanism.
- Reduce the temperature

7.1. Affinity Based Cloning Method

The original B-Cell Algorithm employs a fixed clone size for each solution $\boldsymbol{\gamma}_p$ of the population, preferably equal to the size of the population. In the multi-objective version of the algorithm developed in this paper, each solution is set to be cloned proportional to its fitness. In other words, the clone size for each solution in this algorithm is not fixed. In order to control the computational complexity of the algorithm, in each cycle the sum of the number of cloned solution size for each population solution $\boldsymbol{\gamma}_p$ is set to be a fixed value, say *Nc*. The clone size for each solution is defined such that the fixed total clone size of *Nc* is achieved. To capture the bi-objective nature of the algorithm, the non-domination levels of the solutions are used as a parameter to define the individual clone sizes. The non-domination level of solution $\boldsymbol{\gamma}_p$ which is denoted by $Rank(\boldsymbol{\gamma}_p)\epsilon[1, Rank^{max}]$, where 1 and $Rank^{\max}$ are the highest and the lowest non-domination levels in the population. To compute the clone size $Clone(\boldsymbol{\gamma}_p)$ for each solution $\boldsymbol{\gamma}_p$, first, a weight $CW(\boldsymbol{\gamma}_p)$ is obtained for each solution $\boldsymbol{\gamma}_p$ according to its position among the non-domination levels using equation (34), in which lower $Rank(\boldsymbol{\gamma}_p)$ values result in higher weights.

$$CW(\boldsymbol{\gamma}_p) = \sum_{w=Rank(\boldsymbol{\gamma}_p)}^{Rank^{max}} (w) \Big/ \sum_{w=1}^{Rank^{max}} (w) \tag{34}$$

Next, the following formula is used to obtain the clone size of solution $\boldsymbol{\gamma}_p$ using the normalized weight values, such that $\sum_{w=1}^{P} Clone(\boldsymbol{\gamma}_w) \cong Nc$.

$$Clone(\boldsymbol{\gamma}_p) = Nc \times \frac{CW(\boldsymbol{\gamma}_p)}{\sum_{w=1}^{P} CW(\boldsymbol{\gamma}_w)} \tag{35}$$

Finally, the obtained clone sizes are rounded to their nearest integers. Our preliminary results indicated that using the preceding method to determine clone size results in better algorithmic performance than using the fixed clone size.

7.2. Contiguous Somatic Hypermutation Operator

As mentioned in above section, contiguous somatic hypermutation is a unique feature of the B-cell algorithm. The operator devised for the mBCA uses two methods to generate neighbor solutions, namely the perturbation in lane allocations and the perturbation in expansion or construction projects. The lane allocation perturbation is performed on randomly selected contiguous links (or equivalently a contiguous region according to the BCA terminology), rather than randomly selected individual links. The project perturbation is performed by omitting an existing project and the inclusion of another project in the same design year.

In order to select contiguous links, according to the notions originally defined in the B-cell algorithm, it is necessary to define the hotspot and the so-called the length of the contiguous regions. In the original BCA, the hotspot and length of the contiguous region are correspondingly defined as a random component chosen from the solution vector and a random number of adjacent components selected from the vector with the first component to be the hotspot. In this study, due to the network structure of the solutions, these concepts are defined in a different way. Here, the hotspot is a random node on the network and the contiguous region is the set of adjacent links around the selected random node. The concept used in the crossover operator of NSGA-II properly complies with the above definition. The counts that are assigned to each link corresponding to the selected pivot node can be used to determine the region of connected links. A random count value indicates the extent to which the adjacent links around a random node are to be chosen, which simply translates into the length of the contiguous region. Thus, in a single network, one can randomly define a pivot node and randomly define the size (i.e., length) of the region of links connected to it by selecting a random link count.

Based on the above explanation, the two-stage hypermutation operator is designed as follows:

Phase1 (Lane allocation perturbation): for each design year, randomly decide to apply hypermutation. If the answer is yes, repeat the following steps for each peak's network:

- Randomly select a node in the network.
- Find the link counts, considering the selected node as the pivot node.
- Select a random link count number between 1 and the maximum link count number.
- Choose the links with counts between 1 and the selected count number.
- Perform lane allocation perturbation on the selected links.
  - Compute the feasibility interval for lane allocations of the link (as in Section 6.4).
  - If the feasibility interval is not null, apply a random lane allocation perturbation according to it.

Phase 2 (Project perturbation):

- Check every possible swap of lane addition or link construction projects; i.e., omission of an existing project and inclusion of another project at the same design year.
- If there is at least one feasible swap in terms of the budget level and the network connectivity, select a random swap and apply it.

Any infeasible solution resulting in a disconnected network is discarded and the solution generation procedure is repeated again.

7.3. Evolution Mechanism

In the original version of the BCA, a mutated clone solution substitutes the associated solution in the population if the mutated solution is better in terms of its fitness value. The multi-objective nature of the mBCA implies the use of an evolution strategy which adopts the concept of the dominance status of the clone solutions with respect to their corresponding solution $\boldsymbol{\gamma}_p$ of the population. The first step is to identify the clones which can be candidates to substitute the solution $\boldsymbol{\gamma}_p$ of the population. The next step is to decide whether to allow the clones to substitute the solution and to decide which of the solutions is most suitable for substitution. The evolution mechanism is designed such that it always accepts the substitution of a dominating or a non-dominated clone with its associated solution $\boldsymbol{\gamma}_p$, and it accepts the substitution of a dominated clone with a probability. This mechanism allows the diversification in the algorithm by accepting the inferior solutions in some cases. This probability is defined using a cooling scheme similar to that of simulated annealing algorithm, so such that the dominated clones are more likely to be accepted at the beginning and less likely to be accepted at the end of cycles.

The procedure is performed as below:

- Find the Pareto-optimal clones in the set of mutated clones.
- Divide the Pareto-optimal clones into three sub-sets: (a) clones dominating solution $\boldsymbol{\gamma}_p$, (b) clones that are not dominated by solution $\boldsymbol{\gamma}_p$, and (c) clones dominated by solution $\boldsymbol{\gamma}_p$.
- If sub-set (a) is not null, select the solution with the largest distance to $\boldsymbol{\gamma}_p$ to substitute it.

  Otherwise if sub-set (b) is not null, randomly select a solution to substitute $\boldsymbol{\gamma}_p$.

  Otherwise, select a solution from (c) with the smallest distance to $\boldsymbol{\gamma}_p$ to substitute it with the transition probability $e^{-\Delta/Temp}$.

The distance between solution $\boldsymbol{\gamma}_p$ and a clone solution $\boldsymbol{\gamma}_q$, $Dist(\boldsymbol{\gamma}_p, \boldsymbol{\gamma}_q)$ is computed using relation (26), where Z_m^{max} and Z_m^{min} are the maximum and minimum values of the *m*th objective function in the generated clone. The transition criterion Δ is based on the average cost criterion concept in the Multi-Objective Simulated Annealing (MOSA) (e.g., Nam and Park, 2000). In this study, a modified form of this criterion is adopted, which employs the normalized distances between the objective function values as in relation (36).

$$\Delta = \frac{1}{|M|} * \sum_{m=1,..,M} \frac{\left|Z_m(\boldsymbol{\gamma}_p) - Z_m(\boldsymbol{\gamma}_q)\right|}{Z_m^{max} - Z_m^{min}} \quad (36)$$

The temperature value in the probability function decreases in each cycle of the algorithm, such that the probability of accepting the dominated solutions is reduced gradually. The same temperature level is used for each $\boldsymbol{\gamma}_p$ solution in a single cycle.

8. Computational Results

8.1. Test Problems and Data

The problem addressed in this paper has a number of features similar to the previous published works (i.e., Miandoabchi *et al*., 2013). Therefore, we have used the test problems adopted in those works and have customized or added the required attributes for the problem under this study. The examples include three small, three medium, and one large network as shown in Table 5. The test networks have been adopted from the previous literature and have been modified to include the necessary parameters for the DNDP under the study in that paper. The figures of the test networks are illustrated in the Appendix.

Table 5. Testing networks

Network Size	Network adopted	Notation	No. of nodes	No. of links	No. 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	38	528
Large	The Nagurney (1984) network	NA3	40	66	6

The length of the planning horizon is fixed to 3 years for all problems (i.e., T = 3). For all networks, all links are assumed to be two-way in the base network configuration. Construction costs are assumed to have linear functions of the lane numbers. All travel time functions t_{ij} are assumed to have the form of the Bureau of Public Roads (BPR) function, with α equal to 0.15 and β equal to 4. The function form is indicated below where t_{ij}^{0} is the free flow travel time on arc (i, j).

$$t_{ij}^{\tau\omega}\left(x_{ij}^{\tau\omega}, c_{ij}^{\tau\omega}\right) = t_{ij}^{0}\left(1 + \alpha\left(\frac{x_{ij}^{\tau\omega}}{c_{ij}^{\tau\omega}}\right)^{\beta}\right) \tag{37}$$

8.2. Parameter Setting

The parameter values of the two algorithms were set by using a series of experiments and searching for the parameter ranges in similar algorithms from related papers. Moreover, the parameter values were set so that the computational efforts of the two algorithms were as close as possible to each other and the comparison of the solution qualities obtained by them is relatively fair. To achieve this purpose, the computational efforts were considered as the functions of the number of the network design scenarios to be evaluated in each solution. This can be obtained by multiplying the total number of solutions generated by the algorithm by the number of design scenarios in each solution. The total number of generated solutions in turn depends on the values of the algorithm parameters. Table 6 describes the parameter settings and the approximate computational efforts with their explanations.

Table 6. Parameter settings for the algorithms

Algorithm	Main parameters	Approximate computational effort
Modified NSGA-II	*Population size* (*P*)*:* 60 *No. of generations for small examples* (*G*)*:* 1890 *No. of generations for other examples* (*G*)*:*1100 *Mutation rate:* 0.2	$G \times 2N \times 2T$ In each generation, 2*N* solutions are generated, and each solution has 2*T* design scenarios
mBCA	*Population size (P):* 25 *No. of cycles* (*C*)*:* 45, 98, 106, 88, 97, 106, and 176 for HF to NA40 respectively *Total cloning number (Nc):* 18 *Start temperature:* 10 *Stop temperature:* 1	$C \times P \times (Nc+2) \times 2T$ In each cycle, for all *P* solutions, 1 random clone solution and *Nc*+1 mutated clone solutions are generated, and each solution has 2*T* design scenarios

The start and stop temperatures for the mBCA were defined such that they provide a reasonable range of acceptance reduction rates throughout the algorithm. Since the temperature reduction rate for the mBCA depends on the number of cycles *C*, it was obtained for each test problem separately.

The population size, the number of generations for the modified NSGA-II, and the mutation rate for the modified NSGA-II were set by considering the parameter values used in similar papers and carrying out a series of experiments. For the mBCA, the parameters were set by performing extensive experiments with different parameter settings and algorithm attributes so as to ensure that computational effort for the two algorithms were about the same.

8.3. Software and Hardware

All algorithms were coded and run in the Matlab version R2011b without using any of the existing tool boxes. The tests have been carried out using a laptop with a Core i5-2450M @ 2.5GHz CPU and a 6G RAM. Each algorithm ran 10 times for small examples and 5 times for the others.

8.4. Performance Evaluation

In order to evaluate the capability of the algorithms in achieving the optimal or nearly optimal values in reasonable time, four effectiveness measures are used. Three of the measures are used to investigate the quality of the Pareto-optimal sets generated by the algorithms, along with the run time as the fourth measure:

- M_1: The size of the Pareto-optimal set
- M_2: The set coverage measure proposed in Zitzler *et al.* (2000)
- M_3: The diversity measure of the Pareto-optimal set adopted from Deb *et al.* (2002)

M_2 is used for the pair-wise comparison of the algorithms in terms of the fraction of Pareto-optimal solutions obtained by one algorithm that dominates the solutions obtained from another algorithm. The measure can be mathematically written as follows:

$$Coverage(X_i, X_j) = \frac{\left|\{a_j \in X_j; \exists a_i \succcurlyeq a_j\}\right|}{\left|X_j\right|} \tag{38}$$

where $a_i \succcurlyeq a_j$ means that the solution a_i dominates or equal to the solution a_j. Formula (38) is used to calculate the fraction of the solutions in set X_j that is covered by set X_i. In other words, the formula is used to compute the fraction of solutions in set X_j that is dominated by or equal to at least one solution in X_i. It must be noted that $Coverage(X_i,X_j)$ is not necessarily equal to 1-$Coverage(X_j, X_i)$.

M_3 is used to investigate the diversity of Pareto-optimal sets obtained by the algorithms. This measure computes the spread of the Pareto-optimal set members over the solution space. In this paper, a revised form of this measure is used:

$$Diversity(X_i) = \frac{\sum_{i=1,\dots,|X_i|-1} |E_{i,i+1} - \bar{E}|}{|X_i| - 1} \tag{39}$$

where $E_{i,i+1}$ is the normalized Euclidean distance between two consecutive solutions in the Pareto-optimal set defined as formula (40), in which the solutions are sorted by one of the objective function values, and $\bar{E}$ is the average of those distances.

$$E_{i,i+1} = \left(\sum_{m=1,\dots,M} \left(\frac{Z_m^i - Z_m^{i+1}}{Z_m^{max} - Z_m^{max}} \right)^2 \right)^{1/2} \tag{40}$$

Z_m^{max} and Z_m^{min} are the maximum and minimum values of the *m*th objective function in set X_i. If all solutions in X_i are uniformly spread between the two boundary solutions of the set, then the measure becomes zero, because all distances will be equal to the average distance. Thus, a lower value of the measure implies a better diversity among its solutions.

The summary for the average values of the measures are presented in Table 7. The runtimes are reported in minutes. The total number of best values obtained by each algorithm in each measure is counted and put in the second last column. A larger value of the count means that the algorithm performs better in terms of the measure concerned. The last row concludes whether the algorithm is the best performing algorithm under the measure concerned.

Table 7. Summary of computational results

Example	"Modified" NSGA-II				mBCA			
	M_1	M_2	M_3	*Runtime*	M_1	M_2	M_3	*Runtime*
HF	51.8	0.97	0.051	5	2.6	0.00	0.608	9
ND	59.9	1.00	0.029	21	6.1	0.00	0.111	50
SF1	59.9	1.00	0.024	109	10.6	0.00	0.109	229
NA1	60.0	1.00	0.024	97	8.8	0.00	0.134	176
NA2	60.0	0.91	0.026	82	6.0	0.00	0.142	156
SF2	60.0	1.00	0.021	188	9.4	0.00	0.122	657
NA3	60.0	1.00	0.026	261	11.2	0.00	0.075	996
No. of Better Values in the Column	*7*	*7*	*7*	*7*	*0*	*0*	*0*	*0*
Best Performing Algorithm	✓	✓	✓	✓	-	-	-	-

According to the obtained results, the modified NSGA-II clearly outperforms the second algorithm in terms of the Pareto-optimal set size, quality, and diversity of the solutions with a significantly lower runtime. As shown in Table 7, almost all the population members of the modified NSGA-II are Pareto-optimal. Furthermore, all M_1 values are close to 1 for this algorithm which implies that nearly all the Pareto-optimal solutions of the modified NSGA-II dominate or at least are equal to that of the mBCA. However, because true Pareto-optimal solutions for the test networks are not known, it is not possible to investigate the ability and reliability of the algorithms in achieving exact solutions.

Although the computational efforts of the two algorithms were set to be in the same range in terms of the number of generated solutions, the higher runtimes of mBCA is due to the higher computational effort to produce the new solutions using contiguous hypermutation. Indeed, the mBCA performed additional computations for checking feasible project swaps (i.e., the connectivity and budget feasibility) in contiguous hypermutation. In the modified NSGA-II, the solution was generated by crossover and mutation which did not require additional computational effort, except for budget feasibility checking and omitting one of the projects.

In order to evaluate the effect of new proposed density measure and the evolution mechanism on the performance of NSGA-II, the original and modified versions of the algorithm were compared for small test problems with the same number of runs. Table 8 compares the two versions of NSGA-II for measures M_1 to M_3.

Table 8. Comparison of the two versions of NSGA-II

Example	"Modified" NSGA-II			"Original" NSGA-II		
	M_1	M_2	M_3	M_1	M_2	M_3

HF	51.8	0.42	0.051	48.6	0.68	0.063
ND	59.9	0.50	0.029	60.0	0.08	0.046
SF1	59.9	0.39	0.024	59.6	0.15	0.035
No. of Better Values in the Column	***2***	***2***	***3***	***1***	***1***	***0***

As it is observed from the table, the modified version of NSGA-II achieved lower M_3 values in the Pareto-optimal sets in all three problems, and higher M_2 and M_1 values in ND and SF1.

On average, the modified NSGA-II has achieved a 214% better quality, a 29% higher diversity, and a 2% larger set size compared to the original version of the algorithm. Therefore, it can be deduced that using the new density measure and the evolution strategy in NSGA-II yields better results from the quality and diversity aspects of the solutions set. The advantage of the modified NSGA-II is particularly notable in terms of diversity, which was primarily intended to be improved by proposing new features for the algorithm.

9. Conclusions and Future Research Directions

This paper investigated a multi-objective time-dependent discrete network design problem. Four types of decisions are considered in this problem, 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. The model is in fact a new formulation for the combined decision making for tactical and strategic decisions, which is developed based on the time-dependent modeling concept. The model determines the decisions within the planning horizon with multiple years and two daily peak periods in each year. The two objective functions of this model are the total travel time and the total CO emissions.

The model is formulated as a mathematical problem with equilibrium constraints, which is bi-level in nature. Due to the intrinsic complexity and the non-convexity of the model, two multi-objective evolutionary algorithms, namely the improved non-dominated sorting algorithm II (NSGA-II) and a multi-objective novel version of B-cell algorithm, have been developed to solve for good rather than exact solutions. Both algorithms give a set of Pareto-optimal solutions as their outputs. The performance of the two algorithms was tested using a set of test problems taken from the previous studies. The resulted Pareto-optimal sets were compared using three measures: the size of the sets, and the set coverage, and the diversity measures. The results precisely show that the proposed NSGA-II have achieved a better result in all cases with lower run times. In addition, the results indicate that the new feature of the proposed NSGA-II have improved its performance in terms of quality and the diversity of the solutions.

As mentioned earlier, the problem addressed in this paper is NP-hard and it is much more complex than many of the existing conventional or time-dependent NDPs proposed in the literature. Therefore, any extension to this problem will make it even more complex. This increase in complexity may occur either in the upper or lower level problem in various dimensions such as an increase in: the size of the solution space, the number of problem constraints, the number of times to solve the lower level problem for each design scenario, and the complexity of the lower level problem itself. The complexity increases directly and depends on the nature of the extension. If the extension involves other road network designs such as signal settings, turning restrictions at intersections, or parking space allocation, it may lead to a larger solution space and have more constraints. With a larger solution space, the algorithms need to evaluate much more solutions, but at the same time the runtime need to be faster, leading to a trade-off between speed and quality; otherwise the computational burden will be inhibitive in solving large or even medium networks. On the other hand, more constraints imply more difficulties in building feasible solutions, which may bring additional computational burden to the algorithms. Extensions relating to the transit (i.e., bus, metro, etc.) network design decisions involve another dimension of complexity in addition to the aforementioned issues. This makes the problem multi-modal that requires the mode-split/ traffic assignment problem as the lower level problem, which is more complex than the simple traffic assignment problem. Extensions relating to time intervals of the decisions also affect the number of times required solving the lower level problem. Besides, many of the above extensions require more complex solution encoding (e.g., chromosomes with many layers or multi-part chromosomes) and algorithm operators.

Therefore, any extension to the time-dependent network design problem must be proposed with the consideration of the complexity of the resulting problem. The NP-hard nature of the problem implies exponential growth in the computational effort with the size of the problem. Thus, even the fastest computers may not be able to solve larger (real world sized) sized problems in reasonable time. To tackle this issue, more sophisticated and efficient solution strategies need to be devised. One possible approach is to use parallel metaheuristics to benefit from distributed computing capabilities. Another approach is to find fast and accurate equilibrium traffic flow approximation methods. The latter is a great challenge because common approximation methods such as artificial neural networks seem not to be able to predict the flows or objective function values with desirable accuracy for NDPs.

Acknowledgements

The research was jointly supported by grant (201211159009) from the University Research

Committee of the University of Hong Kong, and a grant from National Natural Science Foundation of China (71271183). The authors are very grateful to the three reviewers for their constructive comments.

References

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.

Deb K., Pratap A., Agarwal S., Meyariavan T. (2002) "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation,* 6(2):182-197.

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.

Farahani R.Z., Miandoabchi E., Szeto W.Y., Rashidi H. (2013) "A Review of Urban Transportation Network Design Problems", *European Journal of Operational Research* 229(1):281-302.

Friesz T.L., Anandalingam G., MehtaN.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.

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.

Kelsey J., Timmis J. (2003) "Immune Inspired Somatic Contiguous Hypermutation for Function Optimisation", *Proceedings, Part I Genetic and Evolutionary Computation Conference (GECCO 2003)*.

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.

Lo H.K., Szeto W.Y. (2004) "Chapter 9: Planning transport network improvements over time". In D. Boyce, & D. H. Lee (Eds.), *Urban and Regional Transportation Modeling: Essays in Honor of David Boyce".* (pp.157-176). Northampton (Mass.): E. Elgar, cop.: Cheltenham (G.B.).

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

Magnanti T.L., Wong R.T. (1984) "Network Design and Transportation Planning: Models and

Algorithms", *Transportation Science* 18(1):1-55.

Meng Q., Yang H. (2002) "Benefit Distribution and Equity in Road Network Design", *Transportation Research Part B* 36(1):19-35.

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

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., Daneshzand F., Szeto W.Y., Farahani R.Z. (2013) "Multi-objective discrete urban road network design", *Computers & Operations Research* 40:2429-2449.

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

Nam D., Park C.H. (2000) "Multiobjective Simulated Annealing: A Comparative Study to Evolutionary Algorithms", *International Journal of Fuzzy Systems* 2(2):87-97.

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

O'Brien L., Szeto W.Y. (2007) "The Discrete Network Design Problem over Time", *HKIE Transactions* 14(4):47-55.

Rilett L.R., Benedek C.M. (1994) "Traffic Assignment under Environmental and Equity Objective", *Transportation Research Record* 1443:92-99.

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

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., Jaber X.Q., Wong, S.C. (2012) "Road Network Equilibrium Approaches to Environmental Sustainability", *Transport Reviews*, 32, 491-518.

Wei C.H., Schonfeld P.M. (1993) "An Artificial Neural Network Approach for Evaluating Transportation Network Improvements", *Journal of Advanced Transportation* 27(2):129-151.

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

Yin Y., Lawphongpanich S. (2006) "Internalizing Emission Externality on Road Networks", *Transportation Research Part D* 11(4):292-301.

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

Appendix

The network topologies of the test problems are as follows:

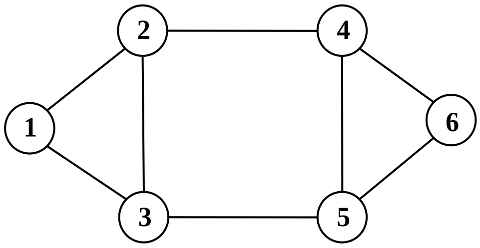

Figure a. Test network HF

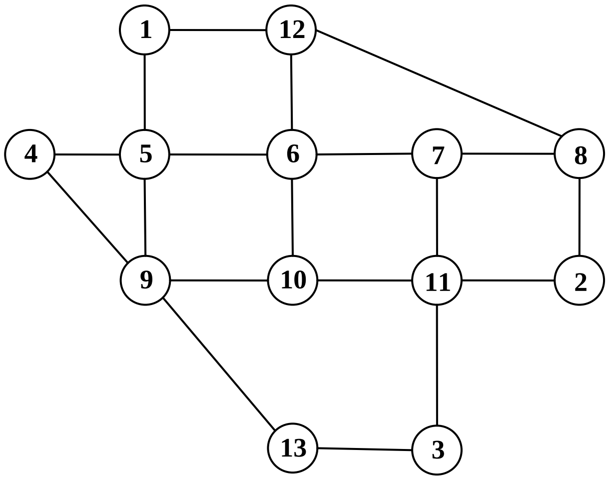

Figure b. Test network ND

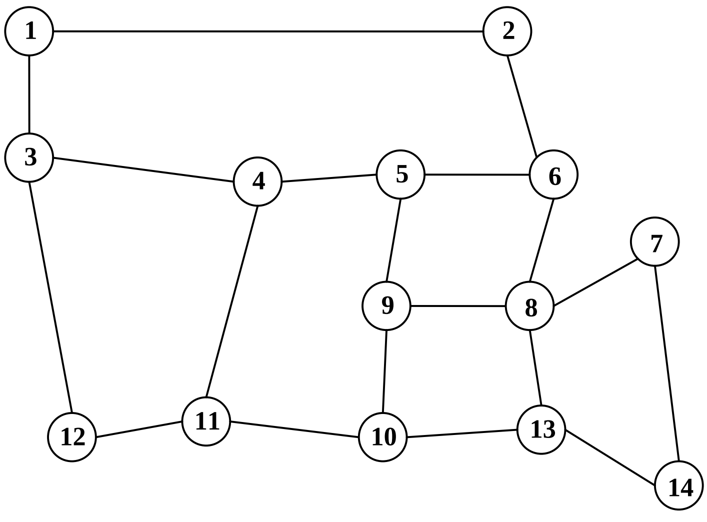

Figure c. Test network SF1

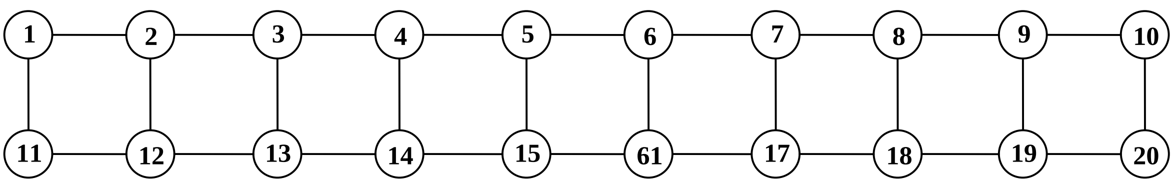

Figure d. Test network NA1

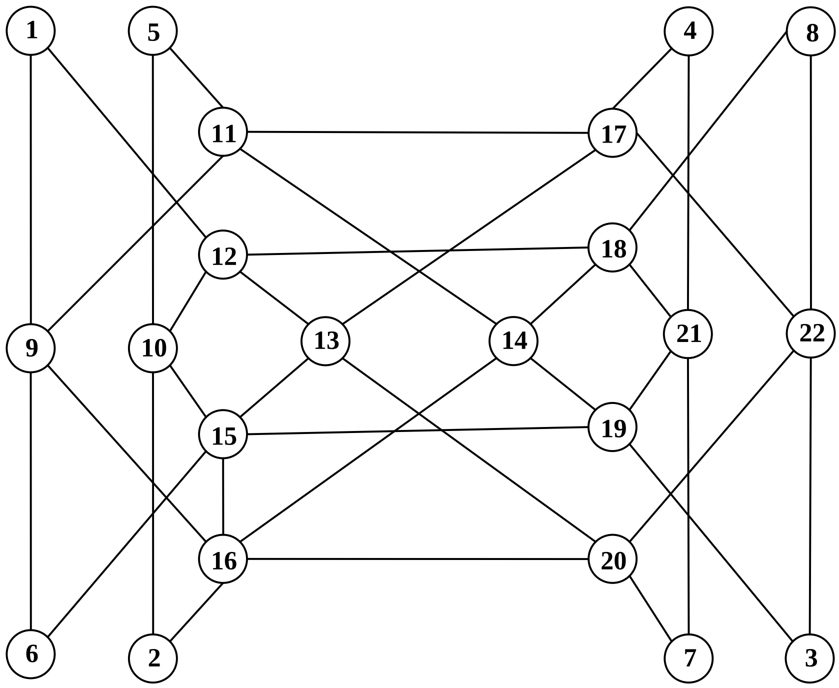

Figure e. Test network NA2

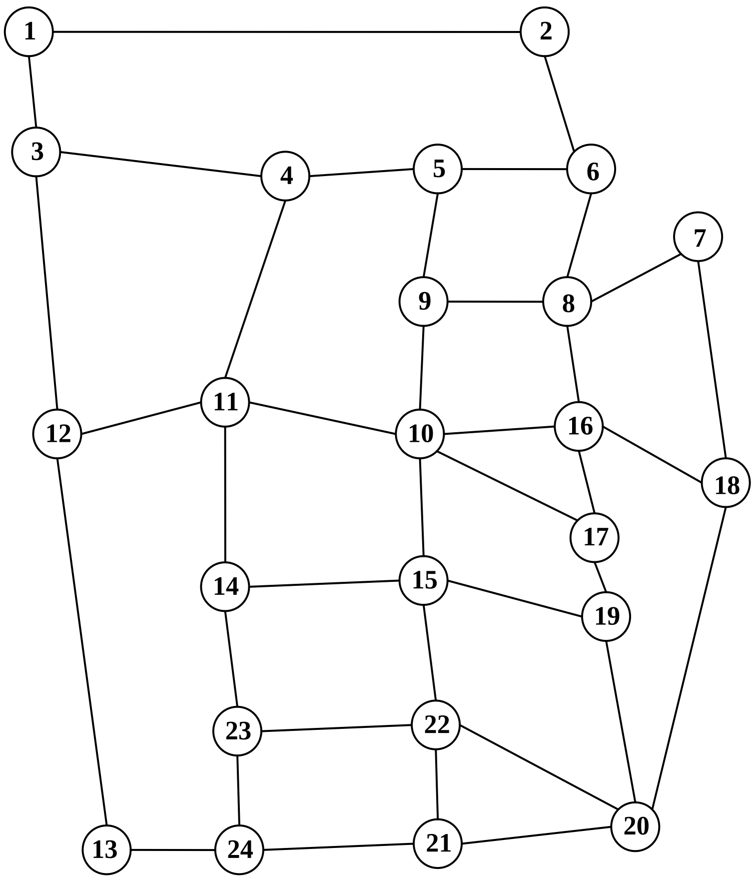

Figure f. Test network SF2