A hybrid meta-heuristic algorithm for solving real-life transportation network design problems

Saeed Asadi Bagloee

Parsons,
2nd Floor, Gulf Tower b, Oud Metha Rd.,
P.O. Box 9123, Dubai, UAE
Fax: +971-4-336-7920
E-mail: saeed.asadi@parsons.com

Madjid Tavana*

Business Systems and Analytics,
La Salle University,
Philadelphia, PA 19141, USA
Fax: +1-267-295-2854
E-mail: tavana@lasalle.edu
*Corresponding author

Avishai Ceder

Department of Civil and Environmental Engineering,
University of Auckland,
20 Symonds Street, Auckland 1142, New Zealand
Fax: 64-9-3652808
E-mail: a.ceder@auckland.ac.nz

Claire Bozic

Chicago Metropolitan Agency for Planning,
233 S. Wacker Drive Suite 800,
Chicago IL 60606, USA
E-mail: cbozic@cmap.illinois.gov

Mohsen Asadi

Kharazmi University,
P.O. Box 15614, Tehran, Iran,
Fax: +98-21-888-830-856
E-mail: mohsen.asadi@tmu.ac.ir

Abstract: The network-design problem (NDP) has a wide range of applications in transportation, telecommunications, and logistics. The idea is to efficiently design a network of links (roads, optical fibres, etc.) enabling the flow of commodities (drivers, data packets, etc.) to satisfy demand characteristics. Various exact and heuristic methods such as branch and bound, Tabu search, genetic algorithm (GA), ant system (AS) have been developed to address the NDP which is a highly intractable combinatorial problem. The literature has yet to address the NDP in real-size networks. In this study, we propose a new meta-heuristic algorithm for solving large NDPs by hybridising GA and AS methods. The applicability of the proposed meta-heuristic approach to real-size networks is demonstrated at two different sites. First, we use a large real-life problem for the city of Winnipeg, Canada and show that our heuristic method produces exact solutions very efficiently. Second, we evaluate the performance of the proposed algorithm using the data of Sioux Falls (a benchmark in the literature). While the proposed approach produces solutions similar to the other available methods in the literature, it is superior for developing solutions in large-size NDPs.

Keywords: hybrid meta-heuristic; transportation; network design problem; NDP; ant system; AS; genetic algorithm; GA.

Reference to this paper should be made as follows: Bagloee, S.A., Tavana, M., Ceder, A., Bozic, C. and Asadi, M. (2013) 'A hybrid meta-heuristic algorithm for solving real-life transportation network design problems', *Int. J. Logistics Systems and Management*, Vol. 16, No. 1, pp.41–66.

Biographical notes: Saeed Asadi Bagloee is a Senior Transportation Planner and Lead Modeller at Parsons in Dubai, UAE. He received his BS in Civil Engineering from Khaje Nasir Toosi University of Technology and his MS in Transportation Planning and Engineering from Sharif University of Technology in Tehran, Iran. His research interests are in transportation planning, demand forecasting, network modeling, transit planning, mathematical programming, econometric analysis, taxation, budget and monetary planning. He has published his research in several scholarly journals.

Madjid Tavana is a Professor of Business Systems and Analytics and the Lindback Distinguished Chair of Information Systems and Decision Sciences at La Salle University where he served as Chairman of the Management Department and Director of the Center for Technology and Management. He has been a Distinguished NASA Research Fellow at Kennedy Space Center, Johnson Space Center, Naval Research Laboratory – Stennis Space Center, and Air Force Research Laboratory. He was recently honoured with the prestigious Space Act Award by NASA. He holds an MBA, a PMIS, and a PhD in Management Information Systems and received his post-doctoral Diploma in Strategic Information Systems from the Wharton School of the University of Pennsylvania. He is the Editor-in-Chief for *Decision Analytics*, the *International Journal of Strategic Decision Sciences*, the *International Journal of Enterprise Information Systems*, and the *International Journal of Applied Decision Sciences*. He has published over 100 research papers in academic journals such as *Decision Sciences*, *Information Systems*, *Interfaces*, *Annals of Operations Research*, *Omega*, *Information and Management*, *Expert Systems with Applications*, *European Journal of Operational Research*, *Journal of the Operational Research Society*, *Computers and Operations Research*, *Knowledge Management Research and Practice*, *Computers and Industrial Engineering*, *Applied Soft Computing*, *Journal of Advanced Manufacturing Technology*, and *Advances in Engineering Software*, among others.

Avishai (Avi) Ceder is a Professor and Chair of Transportation at the Department of Civil and Environmental Engineering and the Director of the newly established Transportation Research Centre (TRC) at the University of Auckland. He was the Head of the Transportation Engineering and Geo-Information Department at the Technion, Visiting Professor at MIT and the University of California at Berkeley. He has published a book entitled *Public Transit Planning and Operation: Theory, Modelling and Practice*, by Elsevier, Oxford, UK. This book was translated to Chinese by the Tsinghua Publishing House, Beijing, China.

Claire Bozic is a Senior Analyst at the Research and Analysis Division of the Chicago Metropolitan Agency for Planning. Her work includes transportation modelling to support major capital project studies, providing expertise for project technical studies, and she can often be found explaining transportation modelling to interested citizens' groups. She chairs the Northeastern Illinois Regional Transportation Operations Coalition and is active in the Northeastern Illinois Advanced Technology Task Force (ITS). She graduated from the University of Illinois in Chicago with a Master of Urban Planning and Policy degree, specialising in transportation.

Mohsen Asadi is a graduate student at the Department of Civil and Environment Engineering, Kharazmi University in Tehran Iran. He received his BS in Civil Engineering from Valiasr University, Rafsanjan, Iran. His research interests are in transportation, environment and computer science. He has presented and published his research in several international journals and proceeding including the *Journal of Transportation Research Records*, *Journal of Transportation Management*, *Proceedings of the 4th International Conference in Sustainable Automotive Technologies* in Melbourne, Australia and the *TRB Annual Meeting* in Washington DC, USA.

1 Introduction

One way to mitigate traffic congestion is to expand the road network by adding lanes to existing roads, adding new roads, and improving intersections. The goal is to identify a set of projects, out of many candidate projects, that provide the most benefit for the lowest cost by minimising the total travel cost subject to a fiscal constraint. This problem, known as the road network-design problem (NDP), has been extensively studied in literature (Boyce, 1984, 2007; Yang and Bell, 1998; Karoonsoontawong and Waller, 2005; Lou et al., 2009). Given a set of candidate projects and a fixed budget, the problem is to find a subset of projects to yield the maximum benefit.

In this study, the travel demand is assumed to be deterministic and fixed, thus the objective is to reduce total travel time. Generally speaking, the total travel time may represent a generalised cost for road users including travel time, toll, safety, and so on.

Let:

$G(V, L)$ a network G, with V and L, representing nodes and links, respectively

(i, j) a directional link belongs to starting from node and it ends at node

L_y set of candidate project links

N number of candidate project links ($N = |L_y|$)

Y vector of project decisions y_{ij}, with $y_{ij} = 1$ if the project link (i, j) is accepted and 0 if the project link is rejected.

L_{y1} set of project links which are accepted in Y: $L_{y1} = \{(i, j) \in L_y \mid y_{ij} = 1\}$

c_{ij} construction cost of project link $\{(i, j) \in L_y$

B budget

x_{ij} traffic flow in link $(i, j) \in L$

$t_{ij}(x_{ij})$ travel time function of link (i, j), it is an increasingly, differentiable, and convex function

$T(L)$ total travel time on network $G(V, L)$ is:

$$T(L) = \sum_{(i,j)\in L} x_{ij} . t_{ij}(x_{ij}) \tag{1}$$

The NDP formulation may be stated as:

$$(NDP) \ \underset{yij}{Max}\, Z\left(y_{ij}\right) = T(L) - T\left(L \cup L_y\right) \tag{2}$$

$$S.t.: \quad \sum_{(i,j)\in L_y} c_{ij} . y_{ij} \le B$$

$$y_{ij} = 1/0, \quad (i, j) \in L_y$$

$$y_{ij} = 1/0, \quad (i, j) \in L_y$$

$$\hat{x}_{ij} \ge 0, \ \textit{is of user equilibrium} (UE) \ \textit{flow}$$

where $\hat{x}_{ij}$ may be the solution of another mathematical-programming problem yielding a pattern of balanced flow known as user equilibrium (UE), where each driver chooses the shortest path (Sheffi, 1985). The NDP is a well-known bi-level programming problem which is classified as an NP hard combinatorial non-linear integer programming problem. The NP hard term refers to problems for which there are no efficient solution algorithms with polynomial computational time. These are notoriously difficult to solve (Jeon et al., 2006; Yang and Bell, 1998; Zeng and Mouskos, 1997).

Two features of the above formulation contribute to the difficulty of the problem. First, the decision variables (y_{ij}) are discrete and not continuous, meaning that additional capacity is defined in units of one lane (Lou et al., 2009). The presence of the discrete or integer variables makes NDP more difficult to solve because the gradient-based approaches are not applicable (Waller and Ziliaskopoulos, 2001; Lou et al., 2009). Adopting discrete decision variables rather than continuous variables better reflects real world conditions because you either build/widen a road or you do not (Jeon et al., 2006; Boyce and Janson, 1980). Second, drivers who response to network changes, are well considered by UE traffic flow which makes the NDP a non-convex problem. An easier, but less accurate, approach is to adopt a system-optimal (SO) traffic flow solution. To calculate the SO flow, the users' shortest path choice behaviour is ignored and instead an artificial flow minimising total system cost is identified (Boyce, 1979; Sheffi, 1985; Waller and Ziliaskopoulos, 2001; Duthie and Waller, 2008).

There are two general approaches to trading-off accuracy and speed for finding solutions. The first approach is the classical branch-and-bound methodology which gives the exact solution (LebLanc, 1975). As the network size becomes significant, however, this methodology is computationally prohibitive. Therefore, many researchers attempted a second approach using non-exact or heuristic methods. There is a wide spectrum of methods in this category, which try to find a satisfactory solution in efficient time (Meng et al., 2004; Jeon et al., 2006). Traditionally this is conducted by relaxing some constraints (Dantzig et al., 1979) or – the objective function (Boyce et al., 1973; Poorzahedy and Turnquist, 1982), shrinking the size of the problem (Haghani and Daskin, 1983), or taking a permuted (reshaped) network (Solanki et al., 1998). Recently, some meta-heuristic search algorithms such as simulated annealing (SA) (Friesz et al., 1993) genetic algorithm (GA) (Yin, 2000; Jeon et al., 2006), ant system (AS) (Poorzahedy and Abulghasemi, 2005) and their combinations (Poorzahedy and Rouhani, 2007) have been developed. These meta-heuristic strategies intend to find a satisfactory solution efficiently, by using historical information and the problem's properties. The approach provides an attractive alternative for large scale applications. Nonetheless, few methodologies have been applied to real world networks.

This study aims to solve real world cases by applying a heuristic approach developed by hybridising the properties of two leading algorithms, GA and AS. Data from the city of Winnipeg, Canada is used to evaluate its performance using a real case study. Data from Sioux Falls, South Dakota, is then used to provide a comparison with recent similar research.

The remainder of this paper is organised as follows. Section 2 briefly reviews the relevant literature. In Section 3, meta-heuristic GA and AS algorithms are introduced. Section 4 provides the details of our proposed hybrid algorithm. Section 5 presents the numerical tests. The conclusions of this study and future-research directions are presented in Section 6.

2 Literature review

The NDP has been addressed in many aspects. This section presents a brief review of different features of the NDP found in the literature. Yang and Wang (2002) compared the widely recognised objective function (adopted in this study) with the reserved-capacity maximisation in the context of network design. They set forth the SA as the solution algorithm to tackle a number of small size artificial cases. Chen and Yang (2004) considered equity issues in the NDP. They developed models utilising GA tested on the small size network of Sioux Falls.

Jeon et al. (2006) adopted a revised GA to address the NDP, and also used Sioux Falls data for numerical tests. Ban et al. (2006) addressed the continuous form of the NDP in which the extra capacity is a real number rather than an integer. Even the additional capacity was treated as a continuous value but the application of the methodology to a real network has yet to be addressed. Zhang and Lu (2007) also dealt with a continuous format of NDP in which, unlike Ban et al. (2006), a budget constraint was taken into consideration. A numerical evaluation was applied to an artificial small network with four nodes. Yin and Ieda (2002) addressed the NDP from the perspective of reliability. They theorised that the road network is expanded to improve the drivers' experience in terms of performance reliability, given a limited budget. The NDP is treated as a continuous problem, and application of the methodology to the real world has yet to be investigated.

Duthie and Waller (2008) incorporated environmental justice into the objective function using GA as the solution approach and Sioux Falls as the case-study. Santos et al. (2008) considered some equity criteria in network capacity expansion, but undertook no real numerical tests. Xu et al. (2009) solved the continuous format of the NDP using GA and SA as the solution algorithms; however the budget constraint is ignored and numerical tests are shown on a network with only six nodes. Xu et al. (2009) concluded that the superiority of one solution algorithm over the others (GA or SA) depends on the level of demand; this finding introduces the concept of hybridisation as a way to enrich the solution algorithm, pursued in our study.

Lou et al. (2009) considered demand uncertainty where the problem is written with complimentary conditions; they developed their solution algorithm on the basis of a cutting-plane method which results in the optimum solution under special conditions. As such, the discrete NDP is heuristically relaxed from some constraints; they used the networks of Sioux Falls and Hull, Quebec (741 links) as case studies. Gallo et al. (2010) developed a meta-heuristic methodology to tackle a special NDP in which the best scheme for roads direction and signals phasing is sought; they evaluated their methodology on a mid-size case (36 zones). It is to note that the delay at the intersection approach links depends on the perpendicular link volumes, which is not captured by an ordinary traffic assignment model like the one used by Gallo et al. (2010). Therefore, the reliability and accuracy of their methodology is questionable.

The literature review emphasises and supports the following two points.

- Our adopted version of the NDP, applied to a set of candidate projects and includes a budget constraint, is more realistic and rigorous than the continuous NDP.
- The literature has yet to deliver efficient methodologies that can solve realistic-network cases.

3 Meta-heuristic algorithms

Meta-heuristic methods attempt to solve computational problems, such as the ND-problem, where there is no satisfactory or applicable mathematical alternative. The intent is to bring a level of intelligence to the search for a solution to the problem by hybridising heuristic search-techniques, such as GA and AS, in an efficient manner. Each of these techniques possesses certain specific intelligence. Clearly, a collection of such abilities can drive the search procedure. The procedures are then fed with historical data which is used to guide the movement from the current solution-state to the new one. This search ends by finding a satisfactory and possibly exact solution. In the interest of brevity, relevant meta-heuristic algorithms are concisely reviewed as follows.

3.1 Ant system

The AS is inspired by the foraging behaviour of real ants. Ants explore the area surrounding their nest in a random manner. As soon as an ant finds a source of food, it evaluates the quantity and quality of the food and carries some of it to the nest. During the return trip, the ant deposits a pheromone trail on the ground, which gradually evaporates. Other ants determine their movements by judging the pheromone density on the paths and choose the one with the most pheromone. In this way ants can find satisfactory foods (Dorigo et al., 1996). Taking advantage of past learning, as represented by the pheromone trails, is the key component of the AS and is exploited in this study.

3.2 Genetic algorithm

The GA is based on the biological mechanism of natural selection and genetics. It codes each feasible solution (or scenario) as a chromosome. A chromosome is a string of several units called genes. Genes have binary values (0 or 1) representing the status of candidate projects (1 if the corresponded candidate contributing to the scenario and 0 otherwise). At each trial, a new chromosome is generated by genetic operators (such as selection, crossover, reproduction or mutation) on a set of selected past-generations. Among them, 'mutation,' with a fairly low probability, is applied to prevent the algorithm's stagnation at local optima. Mutation alters one (or more) gene's value in a chromosome from its initial state (Holland, 1975).

3.3 Fundamentals of the proposed algorithm

Consider several candidate projects to be termed 'candidate'. The goal is to identify a subset of the candidates, given a limited budget, so that the objective function [equation (2)] is maximised. The algorithm consists of several successive iterations. During each iteration some candidates are selected to constitute a network scenario to be termed 'scenario'. The scenario is coded as a chromosome with genes representing the candidates (1= selected, 0 = unselected). The scenario is evaluated and its corresponding chromosome is labelled with a value known as the 'fitness value.' The score of each gene within the chromosome, known as a 'point,' is calculated based on the scenario's fitness value. The point value accumulated by each candidate over a number of iterations where chromosomes are developed is called the candidate's visibility, and this information is used to create the next chromosomes. The more visibility a candidate has, the more attractive it is and the more likely it is to be selected. We define a chromosome with a single filled gene as a 'singular-chromosome' and its fitness value as the 'intrinsic-point' of the corresponding candidate or gene.

The structure of the proposed algorithm of this study is shaped by AS. The mutation concept and terminology of GA (such as chromosome and gene) is also utilised. Following is the description of the algorithm. Each ant starts its journey with a certain budget 'in its pocket'. The ant scans for candidates ahead. Those candidates (or genes) which are more visible and affordable are preferred by the ant. The ant randomly selects the most visible ones successively, paying each candidate's construction cost, and returns to its nest when its budget is expended. At the end of the journey, the algorithm evaluates the ant's success. If the resulting composed chromosome is already is in the algorithm's inventory, the ant is forced to repeat its trip to discover a unique chromosome. Otherwise, the algorithm calculates the amount of food collected, labels it to the corresponding chromosome and saves it. The points for each gene in this chromosome are also calculated. The genes' visibilities, measured in points, are updated based on the latest journey and used for the next ant (iteration). This process continues for a pre-specified number of iterations.

The elements of the algorithm developed are described as follows.

- The random nature of the ant's path-finding behaviour is designed to discover the entire solution domain as far as possible. Inspired by SA, early iterations have a high degree of randomness (Kirkpatrick et al., 1983). As the algorithm progresses, this randomness decreases with converging results [see equations (6) to (4) below].
- The random path-finding behaviour may sometimes result in a replicated journey. A replicated journey is permutated by applying the mutation procedure with a narrow probability of 0.1.
- The collected food corresponding to each chromosome is the amount of saved user-travel time, calculated as:

$$Z_s = T_0 - T_s \tag{3}$$

where Z_s is the benefit (food) of scenario s:

T_0 is total travel time in Do-Nothing network scenario [equation (1)]

T_s is total travel time in scenario s [equation (1), while $L := L \cup L_{y1}$, L_{y1} contributed genes in scenario s].

- The intrinsic-point of each gene, (y_{ij}) corresponding to a singular-chromosome is calculated as:

$$I_g = T_0 - T(L \cup \{(i,j)\}) \quad for\ \forall (i,j) \in L_y, \quad g \in 1 \ldots |L_y| \tag{4}$$

where g indicates the candidates, $g = (i, j) \in L_y$.

- P_g^i, points of candidate g in iteration i are calculated as:

$$P_g^i = \begin{cases} \delta_g^i . \sqrt{Z_i . I_g} & Z_i, I_g > 0 \\ 0 & otherwise \end{cases} \tag{5}$$

where δ_g^i, is 1 if gene g belongs to the chromosome of iteration i, or 0 if it does not. The geometric average is used rather than the mean in order to minimise the influence of the situations where weak genes, (low I_g) are contributing to a strong chromosome (high Z_i). Now V_g^i, the visibility of candidate g at iteration i may be computed as:

$$P_g^i = \begin{cases} \delta_g^i . \sqrt{Z_i . I_g} & Z_i, I_g > 0 \\ 0 & otherwise \end{cases}$$

$$V_g^i = \frac{w_1 . M_g^i + w_2 . \left(I_g / \tilde{C}_g \right)}{w_1 + w_2} \Big/ \tilde{C}_g \tag{6.1}$$

$$M_g^i = \frac{\sum_{c=1}^{pb} \left\{ I_g^c . \left(Z_c / \max_s Z_s \right)^2 \right\}}{\sum_{c=1}^{pb} \left(Z_c / \max_s Z_s \right)^2} \tag{6.2}$$

- M_g^i is the weighted average points derived from the p_b highest rated scenarios, representing the contribution of gene g to the strong chromosomes found so far.
- p_b is a parameter which may be identified as follows. We ensure with significant confidence that every candidate project $s \in S$ will appear at least once in the p_b top chromosomes, all else being equal. p_b depends upon how many candidates (out of total N) contribute to the chromosomes as well as what the budget level (B) is with respect to the total candidate projects costs (C). Therefore, one expects the ultimate number of selected candidate projects to be around ($N.B$ / C). The probability of not having a candidate $s \in S$ contributing to one chromosome is ($N - N.B$ / C) / N or $1 - B$ / C, and the probability of having a candidate $s \in S$ contributing to none of the p_b chromosomes is $(1 - B / C)^{p_b}$. Finally, the probability of having a candidate $s \in S$ contributing at least once to the p_b chromosomes is $1 - (1 - B / C)^{p_b}$. For this probability, we require a significant confidence and hence p_b can be identified. In probability theory, significant confidence is regarded as a 0.95 chance or above. So we have:

  $$1 - (1 - B / C)^{pb} = 0.95 \quad (6.3)$$

  For example, for various budget levels of B / C = {0.1, 0.5, 1} we would have p_b = {29, 5, 1}.

  In equation (6.1) w_r, the arguments' weights are defined as:

  $$w_r = f_r . \rho^{(1 - i / i_{\max})}; \; r = 1, 2 \quad (6.4)$$

  where ρ is a uniform random number between 0 and 1, with an expected value 0.5, $i_{\max}$ is the maximum number of iterations, and the power term ($1 - i$ / $i_{\max}$) enforces the SA concept whereby the randomness of the weights is high in the early iteration and it vanishes as the algorithm proceeds, f_r is the weight-factor of each argument as:

  $$f_1 = 1, \quad f_2 = \alpha, \; (0 \le \alpha \le 1) \quad (6.5)$$

  α is a parameter that balances the interaction between those two arguments, and so it must be calibrated. It also regulates the units of the engaged arguments (M_g^i, (I_g / $\tilde{C}_g$)).

Furthermore, the visibility of the genes [equation (6.1)] should be adjusted based on the candidates' construction costs (C_g) to make less costly candidates more visible in an impartial way. This adds the concept of 'Cost-Benefit analysis' to the problem. On the one hand, costs are exogenous parameters which may be affected by a number of construction and civil factors such as the availability of land and materials procurements, etc. On the other hand, from the transportation-flow perspective, it is an additional network capacity that improves mobility, not the cost of the projects. Regardless of the type of roads (expressway, arterial local etc) a simple representation for the additional network capacity may be defined as the additional driving space measured by candidate's area (length * lanes).

Therefore in the visibility perceived by ants [equation (6.1)] the candidates' construction costs are represented by $\tilde{C}_g$ which is an adjusted term based on the candidates' area:

$$\tilde{C}_g = C_g{}^{\beta} \tag{6.6}$$

where $-1 \leq \beta \leq 1$ is a parameter and is defined as statistical correlation between the candidates' costs C and the candidates' areas A: $\beta = r(A, C)$. For instance, all else being equal, $\beta = 1$ implies that candidates' costs are 100% proportional to the candidates' areas. Computational results show that the concept of adjusted cost $(\tilde{C}_g)$ makes the algorithm too vigilant to allocate the budget on different candidates, to get the maximum possible benefit.

Two notes are worth mentioning:

- In defining the weights of the arguments [equation (6.4)], the core concept of SA is exploited. The SA involves the heating and controlled cooling of materials to improve the quality of output. Here, early in the process, the genes can easily move away from their positions and wander randomly to be absorbed by better chromosomes. These early iterations can be interpreted as the higher heat phase. Slow cooling, as executed by steady steps toward $i_{\max}$, provides more opportunities to find configurations with lower internal energy or better objective function (Kirkpatrick et al., 1983).
- A project's visibility consists of two arguments: Several (p_b) of the topmost precedent scenarios and the intrinsic point. Using these arguments, functions as a way to use the algorithm's memory to target the strong chromosomes. Here, the superiority of the proposed algorithm over GA emerges. Contrary to GA, in the breeding stage, the proposed algorithm takes the genes' qualities into account. More visible genes are more likely to be selected.

Then the algorithm moves forward to select the visible candidates to assemble a chromosome. To do so the genes are stochastically sorted by their visibility in descending order and are collected subject to the budget. Inspired by the mutation concept, a fairly narrow random criterion is embedded too:

$$S^i = \left\{ \frac{V_g^i}{\max\limits_g \left(V_g^i\right)} . \rho^{o_g/N} \geq \varepsilon \right\} \tag{7.1}$$

where S_i is the set of eligible-ordered genes for the scenario of iteration i, O_g is the order of genes in the list sorted in descending order based on their visibility.

The trivial inequality restriction (ε) is called the 'visibility threshold' which works to discard weak genes at the bottom of the list. For a set of good and uniform genes, ε is expected to be 0 because there is no weak gene. Therefore, ε is defined based on the genes' intrinsic point (I_g) used as the fitness measurement, and the standard deviation of I_g as the measurement for uniformity:

$$V_g^i \, \varepsilon = \sigma\left(I_g / A_g\right) \Big/ \sum\nolimits_g \left|I_g / A_g\right| \tag{7.2}$$

where $\sigma(I_g / A_g)$ is the standard-deviation of adjusted intrinsic points based on the candidates areas A_g. The dominator makes a dimension-free trivial number.

The random terms ensures that the strong genes (low o_s) are less likely to be rejected:

$$\left(i.e.\ O_g \rightarrow 1;\ \rho^{O_g/N} \rightarrow 1\right) \tag{7.3}$$

Finally, from the eligible and ordered genes which have passed the threshold, those on the top (strong projects) up to the budget level are selected. In the next section we present the proposed method for solving ND problems.

4 Proposed hybrid algorithm

Let:

i current iteration

T_0 total travel time of do-nothing scenario.

4.1 Step 0 – Prepare and initialise

Set the number of iterations ($i_{\max}$): The maximum number of iterations is limited to the linear order of n:

$$i_{\max} = p.N \tag{8}$$

where, parameter $p > 1$ is an integer value which must be calibrated.

Set the iteration number i: = 0.

Recall singular-chromosomes for iterations $g = 1.N$ do: restore singular-chromosome I_g for $i = g$ and set candidates' points $P_g^g := I_g$.

4.2 Step 1 – Calculate visibility

At iteration i compute V_g^i for every gene g according to equation (6).

4.3 Step 2 – Select projects

With respect to equation (7), consider S^i, the sorted list of eligible genes at iteration i. Start from the first record of S^i: Compare the (remaining) budget and the current candidate's cost. If the candidate is within the budget, then add it to chromosome i. Go to next candidate in list S^i and repeat the procedure until the end of the list is reached, or the budget is exceeded.

4.4 Step 3 – Duplication check

Check whether the current generated chromosome has been created before, If yes, then do:

If random-number generator yields $\rho < 0.1$ then:

Mutations – select two filled and unfilled genes randomly and swap their values,

Go back step3,

Otherwise

Repeating – initialise the sorted list of eligible genes S^i, Go back step 2.

End if

Otherwise repeat this step up to 2.*N* times. If this still fails to generate a non-replicated chromosome then go to Step 5

4.5 Step 4 – Calculate benefit, (traffic assignment)

Execute traffic-assignment module, compute Z_i benefit of scenario *i* according to equation (3).

4.6 Step 5 – End the algorithm

i: $i + 1$

If $i < i_{\max}$ then go to step1.

End.

4.7 Hint-1

The *N* first iterations are dedicated to restoring the already calculated intrinsic-points of the chromosomes. So i_a, the number of traffic assignments, is always $i_a = i_{\max} - N$.

5 Case studies

The algorithm's performance was evaluated using data from Winnipeg, Canada. In addition, and in order to compare the algorithm's performance with recent similar research results, the procedure was also tested using data from the Sioux Falls network. To conduct a challenging evaluation, the algorithm is tested under various budget levels, as measured by the ratio of budget to total candidate project costs (*B* / *C*). The accuracy of the algorithm's results is measured as the percent difference in value of the algorithm's solution and the optimal solution, which is called the error index and is presented as Err (%). The optimal solution was found by an exhaustive enumeration prior to starting the algorithm. Finally, since the meta-heuristic algorithm of concern has a stochastic nature, each running-trial is repeated 50 times, and the average results are reported. This approach is consistent with other research findings (Poorzahedy and Abulghasemi, 2005; Poorzahedy and Rouhani, 2007).

The algorithm, written in Visual Basic, is linked to MS-Excel (to get input data), and EMME/2 (for the traffic assignment routine). In order to efficiently handle the large amount of data, the programme is linked to MS-ACCESS as a database. A PC with a 1.86G Hz CPU and 2.00 GB of RAM is used.

5.1 Winnipeg case-study

The data for Winnipeg is in standard EMME/2 (EMME/2, 1999) format. It consists of 154 zones, 903 nodes and 2,975 links, as well as 4,420 non-zero OD pairs with 56,219 trips as travel demand. There are 15 candidate new bidirectional roads with two lanes in each direction, shown in Figure 1. All characteristics except length, which is calculated geometrically using node coordinates, are taken from the attributes of link $(i, j) = (518, 519)$. Construction cost has been randomly assigned as:

$$C_g = 2.\rho.A_g \tag{9}$$

where the candidate projects' lengths are defined as A_g (note that projects' widths are identical).

Figure 1 Winnipeg, Canada, road network and candidate projects

Table 1 Presentation of 15 new two-way roads for Winnipeg case-study

s	*I-node*	*J-node*	*Length (mi)*	*Construction cost (unit of money)*	*Benefit or intrinsic-point, is saved travel time (minute/hr)*
1	551	610	0.87	0.51	10,424.9
2	424	327	0.94	1.43	1,604.4
3	330	325	0.76	1.25	1,679.4
4	428	330	1.02	0.94	1,421.4
5	494	441	1.22	2.18	824.9
6	437	424	0.49	0.66	404.3
7	423	304	0.73	1.07	6,661.8
8	297	1057	0.57	0.51	2,252.1
9	299	1058	0.79	0.80	4,598.3
10	173	829	0.64	0.91	9,998.8
11	420	592	0.45	0.20	1,918.2
12	739	774	0.41	0.11	428.0
13	288	294	1.47	0.65	5,326.0
14	168	784	0.65	0.10	4,138.7
15	335	449	0.41	0.10	2,843.6
Total			11.42	11.42	

This definition of construction cost is designed to represent real situations which impose varied costs on projects regardless of size. Table 1 introduces these 15 candidate projects as well as each one's number of intrinsic-points.

By enumerating all possible permutations (2^{15} = 32,768) the exact solutions under various budget levels are found. The average CPU time for solving one UE flow problem (traffic-assignment) is approximately 8 seconds (relative gap is 0.1%). Accordingly the enumeration took more than three days. Total travel time in 'Do-Nothing' scenario is T_0 = 786,193.0.

- *Calculable parameters:*
  - a the correlation between 'area' and 'cost' is $r(A, C) = 0.357$
  - *b* P_b the number of highest performing chromosomes to compute visibility in equation (6). According to equations (6) to (3) with respect to $N = 15$, across all the budget levels ($0.1 < B / C < 1$) we consider $P_b = 5$
  - c visibility threshold [equation (7)]; $\varepsilon = 0.079$.

  There are two parameters which must be estimated as follows.

- *Calibrating maximum iterations i_{max}* or i_a, ($i_{max} = i_a + N$): A larger of iterations provides a better chance of finding the optimum result. However, the trade-off is increasing the computational time. The appropriate point to stop running the procedure has to satisfy two criteria. First, the assurance of good results, and second, the computer run time must be within a reasonable limit. The algorithm's response over various numbers of assignments (i_a) is analysed for finding the number of iterations that satisfies these conditions. To address equation (8), the algorithm runs from $i_a = 1$ (or $i_{\max} = 16$) up to finding the exact solution (FES) ($i_a = 25$ or $i_{\max} = 40$) for various budget to cost ratios B/C. Table 2 presents the results, and Figure 2(a) depicts the algorithm's accuracy. For instance, after 15 traffic assignments ($i_a = 15$), at all budget levels, the error index is less than 1%. Since a 1% relative gap convergence is recommended to be sufficient (EMME/2, 1999), the capped number of assignments is set at $i_a = 15$ and the number of iterations is set to $i_{\max} = 30$. Therefore, $p = 2$, ($i_{\max} = 2.N$).

Both Table 2 and Figure 2(b) show the frequency of FES over 50 runs. For $i_a = 15$, FES at all B / C ratios is greater than 30 (out of 50). This implies that over 3 runs, one may have confidence that the exact solution will be found.

- *Calibrating the balancing-parameter:* $0 \leq \alpha \leq 1$*:* For various B / C ratios, variations in the algorithm's solutions over a wide range of α ($0 \leq \alpha \leq 1$) are shown in Figure 2(c). The general trend of the result shows that there is a tangible common range over all budgets to cost ratio levels around $\alpha = 0.6$ which gives better results. At this point, the error index is less than 0.8% (for worse case $B / C = 0.6$), and FES is more than 34 (out of 50). Therefore, $\alpha = 0.6$.

5.1.1 Economic interpretation of the optimal solutions

Figure 2(d) illustrates the exact solutions over various B / C ratios and the strings of binary 0/1 values illustrating the chromosome and indicating the selected projects for each one. The latent information in this graph is valuable to the decision-maker for prioritising the projects. Starting from the very low B / C ratio, a gradual increase in the investment level causes the combination of selected projects to change. The continuous presence of certain individual projects at various budget levels makes them entitled to sit in the top of the list of construction order (Poorzahedy and Turnquist, 1982). Moreover the curve fitted to the records, shown in green, has a significant implication. Let us call this the 'investment-curve' A decision-maker may use information about the incremental benefit of various budget levels (i.e., how much more benefit can be realised through one more unit-of-money investment) to fix the budget. A steep slope on the investment-curve is a good point at which to fix the budget (Bagloee and Tavana, 2012).

5.1.2 Sensitivity of the optimal solution

5.1.2.1 Algorithm's sensitivity to $\tilde{C}_g$

Figure 2(e) shows variations of the algorithm's result over three different definition of adjusted cost $\tilde{C}_g$ [equation (6.6)]:

- $\beta = r(A, C)$; the default definition of $\tilde{C}_g$, utilised in the proposed algorithm
- $\beta = 1$, supposing the adjusted cost simply is the projects' cost (i.e., $\tilde{C}_g = C_g$)
- $\beta = 0$, without $\tilde{C}_g$, (i.e., $\tilde{C}_g = 1$).

As shown, the default definition of $\tilde{C}_g$ yields the best results. Simply taking the projects' cost instead of the adjusted cost ($\beta = 1$) makes the results worse and chaotic. In contrast, 'without $\tilde{C}_g$' status yields a mild result. It upholds earlier speculation that: projects' costs are exogenous parameters and they have little to do with transportation characteristics. Thus, adjusting the costs with the project area ($\beta = r(A, C)$) functions as it was intended.

5.1.2.2 Algorithm's sensitivity to ε

Figure 2(f) depicts the effect of embedding the visibility threshold ε to filter the weak genes. As shown, this initiative moderately enhances the algorithm's performance.

5.1.2.3 Algorithm's sensitivity to w_r (concept of SA)

Concerning equation (6.4) two states are tested. First, the random terms of the weights are completely removed (i.e., $w_r = f_r$). By adopting $i_a = 15$, the algorithm is run for the worst cases, which were previously observed (refer to Table 2: $B / C = 0.3, 0.6, 0.8$). The FESs obtained were 45, 21, and 35, respectively. Second, by retaining the random term and removing the concept of gradual cooling (i.e., $w_r = f_r.\rho$) FESs obtained for the worse cases became 46, 28, and 24, respectively. Compared to the corresponding records in Table 2, the results demonstrate that the concept of SA does a better job because the FES rates are 44, 33, and 39, respectively.

Table 2 Winnipeg, performance of the algorithm for no. of assignments (50runs/trial)

B/C	*No. of permutations*	*Value of objective-function*	*No. of assignments*	*FES/50*	*Err %*	*CPU time (minutes)**
0.1	108	20,458	1	*50*	*0.00*	*0.45*
0.2	926	31,923	1	49	0.23	0.82
			3	*50*	*0.00*	*1.30*
0.3	3,633	37,392	1	0	1.45	0.52
			3	16	0.99	1.08
			5	24	0.76	1.62
			7	33	0.49	2.45
			9	31	0.55	3.45
			11	43	0.20	4.55
			13	40	0.29	5.60
			15	*44*	*0.17*	*6.65*
			17	41	0.26	6.82
			19	49	0.03	7.68
			21	49	0.03	8.08
			23	49	0.03	8.42
			25	48	0.06	8.77
0.4	9,025	43,275	1	48	0.11	0.50
			3	*50*	*0.00*	*1.35*
0.5	16,424	45,138	1	43	0.24	0.52
			3	*50*	*0.00*	*1.88*

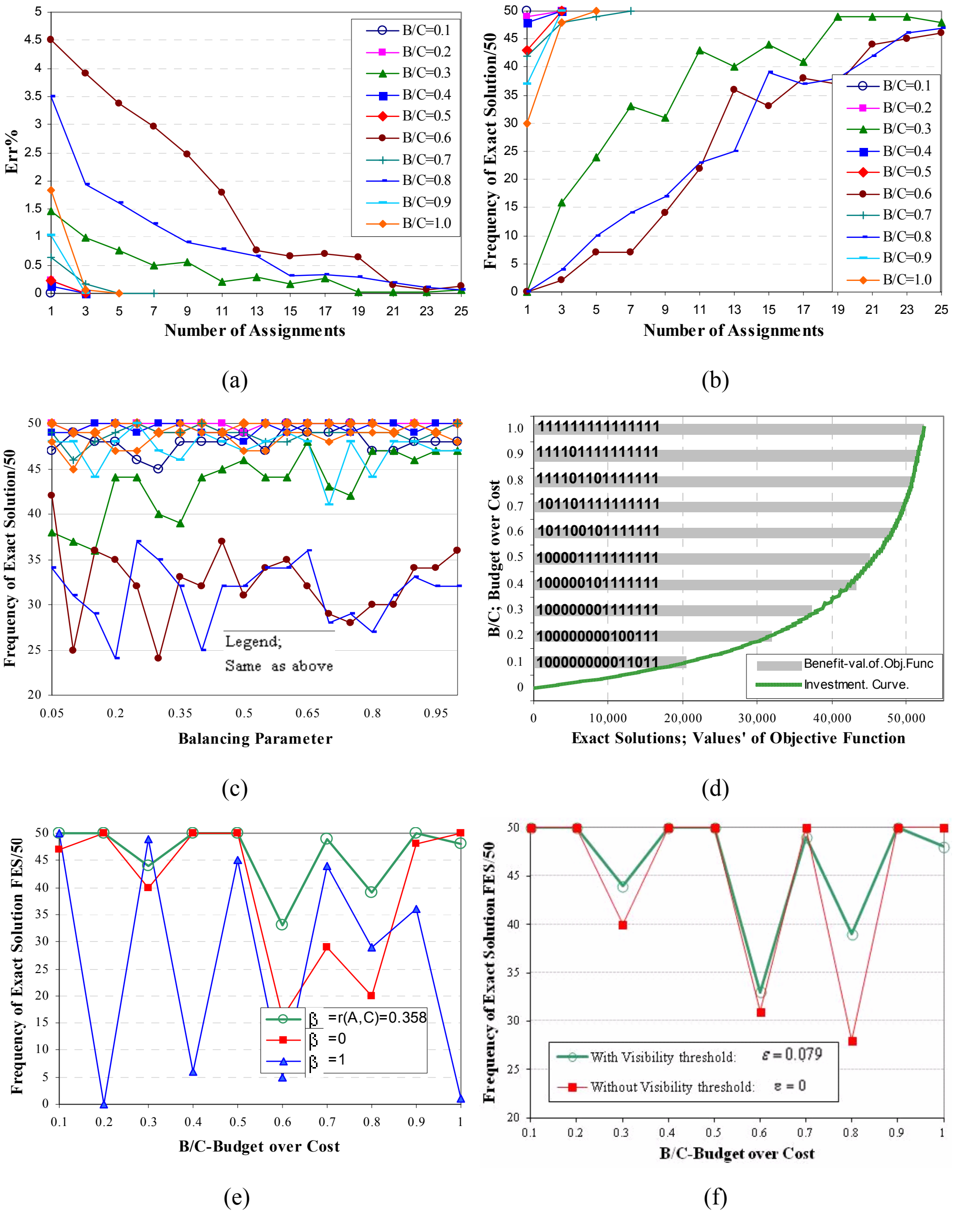
Note: *Running time for 50 runs (in minutes), excluding the traffic-assignment time.

Table 2 Winnipeg, performance of the algorithm for no. of assignments (50runs/trial) (continued)

B/C	*No. of permutations*	*Value of objective-function*	*No. of assignments*	*FES/50*	*Err %*	*CPU time (minutes)**
0.6	23,741	48,015	1	0	4.50	0.55
			3	2	3.90	1.63
			5	7	3.38	2.92
			7	7	2.96	4.10
			9	14	2.46	6.30
			11	22	1.79	6.98
			13	36	0.76	9.67
			15	*33*	*0.66*	*11.05*
			17	38	0.70	12.22
			19	37	0.64	14.32
			21	44	0.14	15.12
			23	45	0.06	15.72
			25	46	0.12	17.43
0.7	29,133	49,681	1	42	0.63	0.55
			3	48	0.17	1.93
			5	*49*	*0.01*	*3.70*
			7	50	0.00	5.25
0.8	31,840	50,806	1	0	3.49	0.55
			3	4	1.94	2.05
			5	10	1.61	3.53
			7	14	1.24	5.42
			9	17	0.90	7.25
			11	23	0.77	9.65
			13	25	0.65	14.45
			15	*39*	*0.30*	*15.70*
			17	37	0.33	18.00
			19	38	0.29	18.93
			21	42	0.19	22.45
			23	46	0.09	22.22
			25	47	0.07	25.20
0.9	32,658	51,421	1	37	1.02	0.57
			3	*50*	*0.00*	*2.12*
1	32,768	52,349	1	30	1.82	0.47
			3	*48*	*0.07*	*1.10*
			5	50	0.00	2.63

Note: *Running time for 50 runs (in minutes), excluding the traffic-assignment time.

Figure 2 Winnipeg case-study, results analysis, (a) specifying no. of iteration, algorithm's convergence, error versus various no. of assignments (iteration) (b) frequency of finding exact solutions versus various number of assignments over 50 runs
(c) specifying balancing-parameter (α), algorithm's sensitive over various ($0 \leq \alpha \leq 1$)
(d) exact solutions over various budget levels, a blueprint for prioritising the projects and investment (e) effect of the adjusted cost $(\tilde{C} = C^{\beta})$ in the visibility index (parameter β) on the algorithm's performance (f) effect of the visibility threshold (ε) on the algorithm's performance (see online version for colours)



In spite of the algorithm's promising performance, there are some major questions still outstanding. The following sections therefore examine:

- What are the special characteristics embedded in the algorithm which make it efficient?
- Where does the performance of this algorithm fall among its constituents (GA, AS) and other similar methods in the literature?
- How does the computational time compare to other methods?

5.2 Sioux Falls case-study

The data used in this study was originally introduced by LebLanc (1975), and used by Poorzahedy and Turnquist (1982), Poorzahedy and Abulghasemi (2005) and Poorzahedy and Rouhani (2007).

The network has 24 zones and 76 links. There are ten pairs of candidate project links. The construction costs of projects 1 – 10 are C_1 = 0.625, C_2 = 0.650, C_3 = 0.850, C_4 = 1.000, C_5 = 1.200, C_6 = 1.500, C_7 = 1.650, C_8 = 1.800, C_9 = 1.950, and C_{10} = 2.100 monetary units. The first five projects are improvements to existing links, and the remaining projects are new links. The total travel time for the 'Do-Nothing' scenario is T_0 = 86,184.

5.2.1 Calculating/calibrating parameters

To calculate the parameters and provide data to the algorithm, the area of the various projects are required. Unfortunately, these are not mentioned in the cited references. Relative values are therefore calculated as follows.

5.2.1.1 Estimating projects' areas

Roads areas are referenced in the functions known as volume delay functions used to calculate travel times. In the Sioux Falls dataset, LebLanc (1975) provided a delay function $t_{ij}(x_{ij}) = a_{ij} + b_{ij}.x_{ij}^4$ where a_{ij}, b_{ij} are parameters. The delay functions follow the format provided by Bureau of Public Roads (BPR) as:

$$\text{(BPR format of road delay function)} \qquad t = t_0.\left(1 + 0.15.(x/Q)^4\right) \tag{10}$$

where t_0 and Q are 'free flow travel time', and 'link capacity', respectively. The t_0 and Q are taken as length and lanes respectively (because, they are in direct proportion). With respect to equation (10) A_{ij} a proxy for the area of road (i, j) can be computed as:

$$A_{ij} = t_0 \times Q_{ij} = a_{ij} \times \sqrt[4]{a_{ij}/b_{ij}} \tag{11}$$

For lane widening project the net added area can be considered as:

$$A_{ij} = \left|A_{ij}^{old} - A_{ij}^{new}\right| \tag{12}$$

where A_{ij}^{old}, A_{ij}^{new} are the proxy areas of the existing links before and after improvement respectively. The required parameters are calculated as follows:

- *Calculable parameters:*
  a $r(A, C) = 0.1626$
  b $P_b = 4$
  c $\varepsilon = 0.055$.
- *Calibrating parameters:*
  a The number of iterations or alternatively, the number of assignments i_a must be calibrated. The number of assignments (trials) affects the algorithm results in both CPU time and solution quality. Therefore, in order to create data comparable to the cited studies (Poorzahedy and Abulghasemi, 2005; Poorzahedy and Rouhani, 2007), i_a is set to the value used in those studies for various budget to cost ratios B / C (see Table 3).
  b The balancing parameter in the visibility index α must be calculated. The average i_a reported in Poorzahedy and Abulghasemi (2005) and Poorzahedy and Rouhani (2007) (at various B / C) is 25. Thus, by assuming $i_a = 25$, the sensitivity analysis to various B / C constitutes a dome shape for α. A simple quadratic regression will provide (e.g., $\alpha = 0.6$ for $B / C = 0.5$):

$$\alpha = -1.85.(B / C)^2 + 1.90.(B / C) + 0.12 \qquad (13)$$

5.2.2 Comparing the algorithm to other approaches

Given these preparations, the proposed hybrid algorithm simply called Gen-Ant is first compared to AS using data from Poorzahedy and Abulghasemi (2005) followed by a comparison with a wide range of hybrid algorithms, developed by Poorzahedy and Rouhani, (2007).

- *Comparison with AS:* Table 3 shows the results of Gen-Ant applied to Sioux Falls, along with AS's results. As shown, AS gives a more sharply accurate result than Gen-Ant. However, the error of AS's solutions in some cases is relatively high, more than 1% and up to 3.5%. It seems that in those cases, AS was unable to cope with climbing the local optimum toward the exact solutions. Although Gen-Ant finds the exact solution less frequently, one may be more confident about the accuracy of the solution provided by Gen-Ant in which the error in the worst case is less than 2%. Compare that to the worst solution provided by AS, with 3.5% error. It seems that contrary to AS in all cases, Gen-Ant is able to climb local optimums toward the exact solutions, albeit slowly and patiently.
- *Comparison with other hybrid algorithms:* Poorzahedy and Rouhani (2007) adopted the AS algorithm as a base-algorithm and improved it by mixing it with some other algorithms including ant colony system (ACS), GA, SA and Tabu search (TS) as follows:

Imp.0	AS, base-algorithm, without improvement
Imp.1	AS + ACS
Imp.2	AS + GA + TS
Imp.3	AS + SA
Imp.4	Imp.2 + Imp.3

Imp.5 Imp.1 + Imp.2

Imp.6 Imp.1 + Imp.3

Imp.7 Imp.1 + Imp.2 + Imp.3.

Table 3 Sioux Falls comparison

Gen-Ant; the proposed algorithm in this study					*The ant system (AS) algorithm, from Poorzahedy (2005), Table 3-(1a)*	
B/C	*Z*	i_a	*FES/50*	*Err %*	*No. of assignments*	*Err %*
0.075	8,024	1	50	0.00	10	0.0
0.150	16,774	9	50	0.00	13	0.0
0.186	20,124	10	50	0.00	12	0.0
0.203	20,124	16	48	0.19	18	0.0
0.235	21,815	18	50	0.00	20	0.0
0.250	24,711	18	50	0.00	20	0.0
0.325	26,052	28	50	0.00	30	0.0
0.375	27,658	29	42	0.25	29	1.1
		42	46	0.09		
0.437	29,266	30	17	1.59	30	3.5
		43	27	0.65		
0.450	29,633	33	44	0.31	33	2.7
		42	46	0.15		
0.460	29,633	30	28	0.06	31	0.0
		44	34	0.01		
0.484	30,530	32	18	1.42	32	2.1
		41	27	0.97		
0.508	30,530	32	4	0.70	32	0.0
		33	12	0.41		
0.531	32,195	29	27	1.93	30	0.1
		39	41	0.68		
0.583	33,287	31	24	1.85	30	0.0
		43	37	0.81		
0.606	33,287	31	28	1.06	31	0.0
		40	37	0.67		
0.625	33,667	28	19	1.34	28	0.0
		37	27	0.70		
0.645	33,761	28	29	0.74	28	0.2
		37	36	0.35		
0.686	34,480	28	25	0.94	27	0.0
		44	45	0.07		
0.702	34,480	25	20	0.90	25	0.0
		42	34	0.31		

Note: *Running time for 50 runs (in minute), excluding the traffic-assignment time.

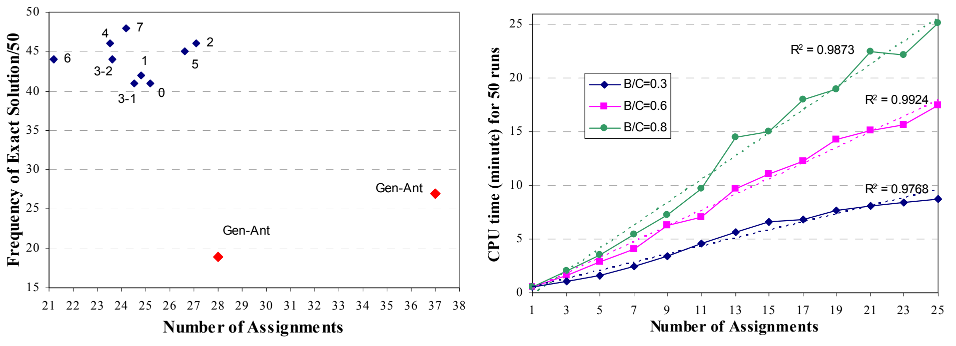
Table 3 Sioux Falls comparison (continued)

Gen-Ant; the proposed algorithm in this study					*The ant system (AS) algorithm, from Poorzahedy (2005), Table 3-(1a)*	
B/C	*Z*	i_a	*FES/50*	*Err %*	*No. of assignments*	*Err %*
0.730	34,788	25	34	0.22	26	0.0
		35	45	0.07		
0.749	34,871	19	13	0.61	20	0.0
		41	27	0.07		
0.812	35,781	18	4	1.84	19	0.0
		44	30	0.63		
0.887	36,368	9	16	0.95	11	0.0
		32	50	0.00		
1.000	86,184	1	37	15.44	1	0.0
		4	49	0.00		

Note: *Running time for 50 runs (in minute), excluding the traffic-assignment time.

Figure 3(a) depicts the results of applying these seven hybrid algorithms, and the AS [Figure 5 from Poorzahedy and Rouhani, (2007)] as well as Gen-Ant to the Sioux Falls network. These results pertain to $B / C = 0.625$ which is more challenging to the Gen-Ant (see Table 3). It shows that the performance of the proposed hybrid algorithm for solving a small size network like Sioux Falls is behind its constituent (AS) and seven hybrid algorithms. Imp.7 showed the best performance. Furthermore Figure 3(a) shows how increasing the number of assignments improves the results of the Gen-Ant.

Figure 3 Accuracy and efficiency of the algorithm (Gen-Ant), (a) Sioux Falls case-study, for $B / C = 0.625$, comparing Gen-Ant to other hybrid algorithms (from Poorzahedy and Rouani, 2007) (b) Winnipeg case-study, efficiency criterion, CPU time is proportional to number of assignments (see online version for colours)



(a) (b)

Imp.7 and AS have been applied to a real network representing the City of Mashhad, Iran (Figure 4). The results showed that Imp.7 generally outperforms the AS algorithm at all budget levels and number of candidate projects. It indicates that the hybrid algorithms perform better than their component parts. In the Mashhad case-study, the size of the search domain is 24,310 permutations while the number of assignments is 97 for algorithm Imp.7, and 78 for AS. In contrast, applying Gen-Ant to the Winnipeg case-study for the worst case (B / C = 0.8) needs only 25 assignments (Table 1) where the size of the search domain is 31,840 permutations. Furthermore, in this study the accuracy criterion has been defined as the difference between found solutions and the exact solution, but for the case-study of Mashhad, the comparison is to the 'best solution found' instead of to the 'exact solution.' Since the 'best solution found' might not be the 'exact solution,' it is unknown what the exact difference is.

Poorzahedy and Rouhani (2007) compared AS to GA for the case-study of Mashhad. The results showed that AS requires less computational time and gives better and much more stable solutions than GA.

5.3 Computational time

Poorzahedy and Turnquist (1982) and Poorzahedy and Rouhani (2007) showed that the CPU times for their algorithm were proportional to the number of assignments. This is a critical point for efficiency concerns. Gen-Ant passes this criterion. Figure 3(b) shows the variation in CPU time for the number of assignments in the Winnipeg case-study (from Table 2). R-square for the fitted linear regression is above 97%. The referred references (Poorzahedy and Abulghasemi, 2005; Poorzahedy and Rouhani, 2007) do not provide enough information about the algorithms' computational time to compare it with the Gen-Ant procedure.

Figure 4 Transportation network; City of Mashhad, Iran

Source: Poorzahedy and Rouhani (2007)

Table 2 indicates that the most time-consuming case happened at B / C = 0.8 and that the number of required assignments was i_a = 25. Thus, B / C = 0.8 and i_a = 25 is the worst case result in terms of computational time. The CPU times reported in Table 2 exclude the assignment times. If the assignment time is included, the total computational time for the worst case takes almost 4 minutes per each run.

6 Conclusions and future research

The NDP has been, and continues to be, an important transportation problem for traffic authorities and researchers. Only a few studies treat the NDP in the context of real-size networks. This study addresses the problem of sizable networks using a meta-heuristic methodology; it develops and proposes a hybrid of leading concepts in search processes, namely, AS, and GA.

The algorithm developed, called Gen-Ant, is evaluated in two case-studies. First, the Gen-Ant algorithm is calibrated and applied to the real-size network for the City of Winnipeg (standard-data of EMME2). The results show that the algorithm can find the exact solutions in an efficient manner in all cases. Second, the proposed methodology is compared with the studies of Poorzahedy and Abulghasemi (2005) and Poorzahedy and Rouhani (2007) using the benchmarked network of Sioux Falls. In the previous two studies, AS is the base and other search methods such as GA, TS and SA have been hybridised to AS. The results of the Sioux Falls benchmark test show that the frequency of exact solutions provided by AS and the hybrid algorithms is higher than Gen-Ant. However, one may be more confident about the solutions provided by Gen-Ant because the error of the worst solution, found by Gen-Ant, is lower.

Moreover, the results using the real Winnipeg network data show that Gen-Ant outperforms the aforementioned algorithms in accuracy and efficiency criteria. Consequently, the developed methodology (Gen-Ant) of this study is better suited to real-life networks than other methods proposed.

The efficiency of Gen-Ant to treat sizable case studies is the outcome of the following elements.

1 This study introduces the concept of 'adjusted cost' [equation (6)]; it adds a layer of reality-seeking behaviour by efficiently selecting candidate projects using the cost of accrued benefits from which the time required for reaching the exact solution is decreased.

2 Contrary to other methods (LebLanc, 1975; Poorzahedy and Turnquist, 1982; Poorzahedy and Abulghasemi, 2005; Poorzahedy and Rouhani, 2007) the number of iterations is specified beforehand; the algorithm using SA [equation 6.4)] is then attained its best solution within a limited number of iterations.

3 A threshold value for selection of candidate projects is defined [visibility threshold ε in equations (7.1, 7.2)] to eliminate weak projects.

4 The performance of the Gen-Ant procedure improves with the size of the problem demonstrating a promising result for real-life network the applications.

Acknowledgements

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions.

References

Bagloee, S.A. and Tavana, M. (2012) 'An efficient hybrid heuristic method for prioritizing large transportation projects with interdependent activities', *International Journal of Logistics Systems and Management*, Vol. 11, No. 1, pp.114–142.

Ban, X., Liu, H.X., Lu, J. and Ferris, M.C. (2006) 'Decomposition scheme for continuous network design problem with asymmetric user equilibria', *Journal of the Transportation Research, Record*, No. 1964, pp.185–192, Transportation Research Board of National Academies, Washington, DC.

Boyce, D. (2007) 'An account of a road network design method: expressway spacing, system configuration and economic evaluation', in Feng, X. and Popescu, A.M. (Eds.): *Infrastructure Problems Under Population Decline*, pp.131–159, Schriften zur öffentlichen Verwaltung und öffentlichen Wirtschaft, Bd. 202, Berliner Wissenschafts-Verlag, Berlin.

Boyce, D.E. (1979) 'Introduction', *Transportation Research B*, Vol. 13, No. 1, pp.1–3.

Boyce, D.E. (1984) 'Urban transportation network equilibrium and design models: recent achievements and future prospectives', *Environment and Planning A*, Vol. 16, No. 11, pp.1445–1474.

Boyce, D.E., Farhi, A. and Weischedel, R. (1973) 'Optimal network design problem: a branch and bound algorithm', *Environment and Planning*, Vol. 5, No. 4, pp.519–533.

Boyce, E.D. and Janson, B.N. (1980) 'A discrete transportation network design problem with combined trip distribution and assignment', *Transportation Research Part B*, Vol. 14, Nos. 1–2, pp.147–154.

Chen, A. and Yang, C. (2004) 'Stochastic transportation network design problem with spatial equity constraint', *Journal of the Transportation Research, Record*, No. 1882, pp.97–104, Transportation Research Board of National Academies, Washington, DC.

Dantzig, G.D., Harvey, R.P., Lansdowne, Z.F., Robinson, D.W. and Maier, S.F. (1979) 'Formulating and solving the network design problem by decomposition', *Transportation Research B*, Vol. 13, No. 1, pp.5–17.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996) 'The ant system: optimization by a colony of cooperating agent', *IEEE Transactions on Systems: Man and Cybernetics, Part B*, Vol. 26, No. 1, pp.1–13.

Duthie, J. and Waller, S.T. (2008) 'Incorporating environmental justice measures into equilibrium-based network design', *Journal of the Transportation Research, Record*, No. 2089, pp.58–65, Transportation Research Board of National Academies, Washington, DC.

EMME/2 (1999) *User's Manual: Software Release 9*, INRO, Montreal, Quebec, Canada.

Friesz, T.L., Anandalingam, G., Mehta, N.J., Nam, K., Shah, S.J. and Tobin, R.L. (1993) 'The multiobjective equilibrium network design problem revisited: simulated annealing approach', *European Journal of Operational Research*, Vol. 65, No. 1, pp.44–57.

Gallo, M., D'Acierno, L. and Montella, B. (2010) 'A meta-heuristic approach for solving the Urban network design problem', *European Journal of Operations Research*, Vol. 201, No. 1, pp.144–157.

Haghani, A.E. and Daskin, M.S. (1983) 'Network design application of an extraction algorithm for network aggregation', *Journal of the Transportation Research, Record*, No. 944, pp.37–46, Transportation Research Board of National Academies, Washington, DC.

Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Jeon, K., Lee, J., Ukkusuri, S. and Waller, S.T. (2006) 'Selectorecombinative genetic algorithm to relax computational complexity of discrete network design problem', *Journal of the Transportation Research, Record*, No. 1964, pp.91–103, Transportation Research Board of National Academies, Washington, DC.

Karoonsoontawong, A. and Waller, S.T. (2005) 'Comparison of system- and user-optimal stochastic dynamic network design models using Monte Carlo bounding techniques', *Journal of the Transportation, Research Record*, No. 1923, pp.91–102, Transportation Research Board of National Academies, Washington, DC.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983) 'Optimization by simulated annealing', *Science*, Vol. 220, No. 4598, pp.671–680.

LebLanc, L.J. (1975) 'An algorithm for discrete network design problem', *Transportation Science*, Vol. 9, No. 3, pp.183–199.

Lou, Y., Yin, Y. and Lawphongpanich, S. (2009) 'A robust approach to discrete network designs with demand uncertainty', *Journal of the Transportation Research, Record*, No. 2090, pp.86–94, Transportation Research Board of the National Academies, Washington, DC.

Meng, Q., Lee, D.H., Yang, H. and Huang, H.J. (2004) 'Transportation network optimization problems with stochastic user equilibrium constraints', *Journal of the Transportation Research, Record*, No. 1882, pp.113–119, Transportation Research Board of the National Academies, Washington, DC.

Poorzahedy, H. and Abulghasemi, F. (2005) 'Application of ant system to network design problem', *Transportation*, Vol. 32, No. 3, pp.251–273.

Poorzahedy, H. and Rouhani, O. (2007) 'Hybrid meta-heuristic algorithms for solving network design problems', *European Journal of Operational Research*, Vol. 182, No. 2, pp.578–596.

Poorzahedy, H. and Turnquist, M.A. (1982) 'Approximate algorithms for the discrete network design problem', *Transportation Research B*, Vol. 16, No. 1, pp.45–56.

Santos, B., Antunes, A. and Miller, E. (2008) 'Integrating equity objectives in a road network design model', *Journal of the Transportation Research, Record*, No. 2089, pp.35–42, Transportation Research Board of the National Academies, Washington, DC.

Sheffi, Y. (1985) *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, New Jersey, Prentice-Hall.

Solanki, R.S., Gorti, J.K. and Southworth, F. (1998) 'Using decomposition in large-scale highway network design with quasi-optimization heuristic', *Transportation Research B*, Vol. 32, No. 2, pp.127–140.

Waller, S.T. and Ziliaskopoulos, A.K. (2001) 'Stochastic dynamic network design problem', *Journal of the Transportation Research Record*, No. 1771, pp.106–113, Transportation Research Board of National Academies, Washington, DC.

Xu, T., Wei, H. and Hu, G. (2009) 'Study on continuous network design problem using simulated annealing and genetic algorithm', *Expert Systems with Applications*, Vol. 2, No. 36, pp.1322–1328.

Yang, H. and Bell, M.G.H. (1998) 'Models and algorithms for road network design: a review and some new developments', *Transport Review*, Vol. 18, No. 3, pp.257–278.

Yang, H. and Wang, J.Y.T. (2002) 'Travel time minimization versus reserve capacity maximization in the network design problem', *Journal of the Transportation Research Record*, No. 1783, pp.17–26, Transportation Research Board of National Academies, Washington, DC.

Yin, Y. (2000) 'Genetic algorithm-based approach for bilevel programming models', *Journal of Transportation Engineering: ASCE*, Vol. 126, No. 2, pp.115–120.

Yin, Y. and Ieda, H. (2002) 'An optimal improvement scheme for network reliability', *Journal of the Transportation Research Record*, No. 1783, pp.1–6, Transportation Research Board of National Academies, Washington, DC.

Zeng, Q. and Mouskos, K.C. (1997) 'Heuristic search strategies to solve transportation network design problem', Tech. Report, Dept. of Transportation and the National Center for Transportation and Industrial Productivity, New Jersey, USA.

Zhang, G. and Lu, J. (2007) 'Genetic algorithm for continuous network design problem', *Journal of Transportation Systems Engineering and Information Technology*, Vol. 7, No. 1, pp.101–105.