European Journal of Operational Research 201 (2010) 144–157

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

A meta-heuristic approach for solving the Urban Network Design Problem

Mariano Gallo [a,*], Luca D'Acierno [b], Bruno Montella [b]

[a] Department of Engineering, University of Sannio, Piazza Roma 21, 82100 Benevento, Italy
[b] Department of Transportation Engineering, 'Federico II' University of Naples, via Claudio 21, 80125 Naples, Italy

ARTICLE INFO

Article history:
Received 11 September 2007
Accepted 15 February 2009
Available online 26 February 2009

Keywords:
Network design
Transportation
Scatter Search

ABSTRACT

This paper proposes an optimisation model and a meta-heuristic algorithm for solving the urban network design problem. The problem consists in optimising the layout of an urban road network by designing directions of existing roads and signal settings at intersections. A non-linear constrained optimisation model for solving this problem is formulated, adopting a bi-level approach in order to reduce the complexity of solution methods and the computation times. A Scatter Search algorithm based on a random descent method is proposed and tested on a real dimension network. Initial results show that the proposed approach allows local optimal solutions to be obtained in reasonable computation times.

1. Introduction

The Urban Network Design Problem (UNDP) consists in optimising the features of an urban road network, without providing infrastructural interventions (design of new roads, widening of existing roads, etc.); it may concern the entire city road network or part of it. In practical applications, the UNDP arises when a policy maker seeks to implement new traffic schemes to improve network performances, reducing the user's total travel time and externalities due to congestion. Since changes in circulation schemes influence traffic flows and conflict points at intersections, the design always needs to consider signal settings as well.

In the UNDP the decision variables, i.e. the variables on which the planner can directly operate, may be classified as follows:

(a) topological variables: directions of links;
(b) signal setting variables: cycle length, effective green times, offsets, phase plan of signalised intersections;
(c) parking variables: parking allowed or not;
(d) fare variables: road and/or parking pricing.

Variables (a) and (c) are discrete while variables (b) and (d) are generally assumed continuous; since the UNDP have to consider at least variables (a) and (b), it is a mixed integer-continuous optimisation problem.

The model proposed in this paper considers only decision variables (a) and (b); variables (c) could be considered without theoretical problems, but only by complicating the model that would require explicit modelling of parking availability and parking search time. Variables (d), instead, cannot be considered in the model without assuming the demand as elastic, at least for the mode choice model; in D'Acierno et al. (2006) the parking pricing optimisation problem is studied in depth. In the model proposed in this paper the demand is assumed inelastic with respect to travel times that change as the network layout changes. Other problem variables are traffic flows; these are descriptive variables that are constrained to decision ones by the assignment procedure.

The objective function to minimise is generally the user's total travel time on the network; other objective functions could be proposed (e.g. minimising pollutant emissions). Some constraints have to be considered in the problem: physical constraints on topological variables (road directions), constraints on signal settings, connectivity constraints, etc.

The UNDP can be seen as a particular case of the more general Supply Design Problem (Cascetta, 2001) or the extensively studied Network Design Problem (Magnanti and Wong, 1984; Feremans et al., 2003).

Some recent papers deal with the UNDP. Cantarella and Vitetta (2006) proposed a heuristic multi-criteria technique based on genetic algorithms, which considered decision variables (a)–(c). Cantarella et al. (2006) proposed and compared some meta-heuristic algorithms for solving the UNDP, namely Hill Climbing, Simulated Annealing, Tabu Search, Genetic Algorithms and Path Relinking, considering decision variables (a) and (b). Both papers proposed to solve the problem by two-stage solution methods. Russo and Vitetta (2006) proposed a method for reducing the number of possible solutions to a UNDP.

The general Network Design Problem is more widely treated in the literature: almost all papers considered only topological variables (a) and were suitable for solving extra-urban problems or

* Corresponding author. Tel.: +39 0824 305565; fax: +39 0824 325246.
E-mail addresses: gallo@unisannio.it (M. Gallo), dacierno@unina.it (L. D'Acierno), montella@unina.it (B. Montella).

0377-2217/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2009.02.026

more general topological network design problems. By limiting the literature review only to models and methods proposed for road networks, the papers can be classified according to the kind of decision variables.

Discrete variable models were formulated in papers by Billheimer and Gray (1973), Los (1979), Boyce and Janson (1980), Poorzahedy and Turnquist (1982), Herrmann et al. (1996), Solanki et al. (1998) and Gao et al. (2005), who proposed *heuristic* solution algorithms, and in papers by Le Blanc (1975), Foulds (1981), Los and Lardinois (1982), Chen and Alfa (1991) and Cruz et al. (1999), who proposed *Branch and Bound* solution algorithms; Drezner and Wesolowsky (2003), Poorzahedy and Abulghasemi (2005), Ukkusuri et al. (2007) and Poorzahedy and Rouhani (2007) constructed some meta-heuristic algorithms.

Continuous variable models were formulated by Dantzig et al. (1979), Marcotte (1983), Harker and Friesz (1984), Le Blanc and Boyce (1986), Suwansirikul et al. (1987) and Meng et al. (2001), who proposed *heuristic* solution algorithms; Abdulaal and Le Blanc (1979), Davis (1994), Cho and Lo (1999) and Chiou (2005) proposed, instead, *descent algorithms*; *simulated annealing* approaches were suggested by Friesz et al. (1992) and Meng and Yang (2002).

The problem of optimising signal settings has been widely studied; it can be seen as a UNDP that considers only the signal setting variables (b). A general framework was reported by Marcotte (1983), Sheffi and Powell (1983), Cantarella et al. (1991), Cantarella and Sforza (1995) and Cascetta et al. (1999). A more detailed literature review can be found in Cascetta et al. (2006). Recently, Teklu et al. (2007) proposed a Genetic Algorithm for optimising traffic control signals.

In this paper, a general non-linear constrained optimisation model for solving the UNDP is formulated. The general model is then modified to allow a bi-level solution approach, reducing the complexity of solution methods and computation times. The proposed model allows the decision variables to be reduced to topological ones alone, since the signal settings are obtained within the equilibrium assignment procedure. This approach is similar to that proposed by Cantarella et al. (2006) and differs from it in the asymmetric assignment model adopted (stochastic instead of deterministic). The asymmetric stochastic user equilibrium algorithm adopted in this paper is proposed in Cascetta et al. (1999, 2006) and will not be reported in this paper. This paper, therefore, focuses mainly on the meta-heuristic algorithm proposed for designing directions, that includes in the method for estimating the equilibrium traffic flows the optimisation of signals with fixed topology. The meta-heuristic algorithm adopted is the Scatter Search (see Laguna, 2002; Glover et al., 2003) that in this paper is adapted to the specific problem to solve. Moreover, for reducing significantly the computation time of the scatter search procedure, a stochastic neighbourhood search method is proposed and compared with the more generally used steepest descent method (or Hill Climbing).

The paper is structured as follows: Section 2 focuses on the model formulation; the solution algorithm is proposed in Section 3; numerical results on a real network are reported in Section 4; conclusions are summarised in Section 5.

2. Optimisation model

2.1. General formulation

The UNDP can be formulated by the following general constrained optimisation model:

$$[\boldsymbol{y}^\wedge, \boldsymbol{g}^\wedge] = \mathrm{Arg}_{\boldsymbol{y},\boldsymbol{g},\boldsymbol{f}^*} \min w(\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{f}^*) \tag{1}$$

subject to:

$$\boldsymbol{y} \in Y, \tag{2}$$

$$\boldsymbol{g} \in G, \tag{3}$$

$$\boldsymbol{f}^* = \boldsymbol{f}^*_{sym}(\boldsymbol{y}, \boldsymbol{g}), \tag{4}$$

where $\boldsymbol{y}$ is the vector of topological variables; $\boldsymbol{g}$ is the vector of signal setting variables; $\boldsymbol{y}^\wedge$ is the optimal solution for $\boldsymbol{y}$; $\boldsymbol{g}^\wedge$ is the optimal solution for $\boldsymbol{g}$; $w(\cdot)$ is the objective function; $\boldsymbol{f}^*$ is the equilibrium flow vector that is obtained by a symmetric equilibrium assignment; Y represents the feasible set for $\boldsymbol{y}$; G represents the feasible set for $\boldsymbol{g}$.

Eqs. (2) and (3) represent the feasibility constraints for decision variables. Eq. (4) represents the assignment constraint that links the equilibrium traffic flow vector to the network layout. Estimation of vector $\boldsymbol{f}^*$ can be obtained (see for instance Cascetta, 2001) by a deterministic (*Deterministic User Equilibrium*, DUE) or stochastic (*Stochastic User Equilibrium*, SUE) approach, according to assumptions on the route choice model adopted. In both cases, given a transportation supply layout (i.e. given the vectors $\boldsymbol{y}$ and $\boldsymbol{g}$), under some assumptions on cost functions, it is possible to prove that the equilibrium flow vector $\boldsymbol{f}^*$ exists and is unique (Cascetta, 2001). Therefore, Eq. (4) is an *application*: to each supply layout, identified by vectors $\boldsymbol{y}$ and $\boldsymbol{g}$, corresponds one and only one symmetric equilibrium link flow vector $\boldsymbol{f}^*$.

The UNDP consists in searching, among all feasible supply layouts $(\boldsymbol{y},\boldsymbol{g})$, for the one $(\boldsymbol{y}^\wedge,\boldsymbol{g}^\wedge)$ which corresponds to the optimal value of the objective function, $w(\cdot)$.

Since Eq. (4) is an application, the vector $\boldsymbol{f}^*$ is univocally determined once $\boldsymbol{y}$ and $\boldsymbol{g}$ are given. Hence the constraint (4) may be included within the objective function:

$$[\boldsymbol{y}^\wedge, \boldsymbol{g}^\wedge] = \mathrm{Arg}_{\boldsymbol{y},\boldsymbol{g}} \min w(\boldsymbol{y}, \boldsymbol{g}, \boldsymbol{f}^*_{sym}(\boldsymbol{y}, \boldsymbol{g})) \tag{5}$$

subject to:

$$\boldsymbol{y} \in Y, \tag{6}$$

$$\boldsymbol{g} \in G. \tag{7}$$

The model thus formulated is a non-linear constrained optimisation model with the following features:

(i) the variables are heterogeneous, since the topological ones are necessarily discrete while the signal settings are, in general, assumed continuous;
(ii) to calculate a value of the objective function an equilibrium assignment has to be performed.

The first feature suggests that a bi-level model can be proposed as follows:

(*Upper level*)

$$\boldsymbol{y}^\wedge = \mathrm{Arg}_{\boldsymbol{y}} \min w(\boldsymbol{y}, \boldsymbol{g}^\wedge, \boldsymbol{f}^*_{sym}(\boldsymbol{y}, \boldsymbol{g}^\wedge)) \tag{8}$$

subject to:

$$\text{(a) } \boldsymbol{y} \in Y. \tag{9}$$

(*Lower level*)

$$\text{(b) } \boldsymbol{g}^\wedge = \mathrm{Arg}_{\boldsymbol{g}} \min z(\boldsymbol{g}, \boldsymbol{f}^*_{sym}(\boldsymbol{y}, \boldsymbol{g})) \tag{10}$$

subject to:

$$\boldsymbol{g} \in G. \tag{11}$$

The lower level is a model for solving the widely studied signal setting problem (see Cascetta et al., 2006). This problem consists in searching for the optimal signal setting once the network topology is fixed. Generally, assuming only the effective green times as variables, it is a non-linear continuous optimisation problem. This problem could be solved with one of the many methods proposed in the literature (Sheffi and Powell, 1983; Heydecker and Khoo, 1990; Yang and Yagar, 1995; Oda et al., 1997; Wong, 1997; Wong and Yang, 1997; Chiou, 1999; Wong et al., 2002; Ziyou and Yifan, 2002; Cascetta et al., 2006; Teklu et al., 2007).

The upper level can be seen as a model for solving the topological network design problem; this problem – a non-linear discrete optimisation problem – consists in searching for the optimal layout of the network topology (road link directions).

The difficulty in solving the bi-level model arises from the necessity of solving the inner level at each iteration of an algorithm that solves the upper level. Hence the computation times for solving the inner level need to be minimised.

To achieve this aim, in this paper we propose solving the signal setting problem with a local approach (Cantarella et al., 1991), which also follows the approach proposed by Cantarella et al. (2006). Accordingly, the signal setting problem may be formulated as an asymmetric equilibrium assignment problem (Cascetta et al., 1999, 2006):

$$\boldsymbol{f}^* = \boldsymbol{f}_{SNL}(\boldsymbol{c}^*, \boldsymbol{y}),$$
$$\boldsymbol{c}^* = \boldsymbol{c}(\boldsymbol{f}^*, \boldsymbol{g}^*),$$
$$\boldsymbol{g}^* = \boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*),$$

$$\boldsymbol{f}^* = \boldsymbol{f}_{SNL}(\boldsymbol{c}(\boldsymbol{f}^*, \boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*)), \boldsymbol{y}), \tag{12}$$

where $\boldsymbol{f}_{SNL}(\boldsymbol{c}^*, \boldsymbol{y})$ is the stochastic network loading function; $\boldsymbol{c}(\boldsymbol{f}^*, \boldsymbol{g}^*)$ is the link cost function vector; $\boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*)$ is the local optimal signal setting vector, obtained applying a local control policy to all isolated intersections.

In solving the fixed-point problem (12), the equilibrium flows, $\boldsymbol{f}^*$, and the local optimal signal settings, $\boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*)$, are obtained.

Cascetta et al. (1999, 2006) proved the existence of a solution of the fixed-point problem (12) and proposed efficient algorithms for finding it, under the assumption of a stochastic route choice model; even if the uniqueness is not stated, the proposed algorithms give a fixed-point solution, $(\boldsymbol{f}^*, \boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*))$, for each layout of topological variables, $\boldsymbol{y}$. Therefore, the relation (12) can be written as:

$$\boldsymbol{f}^* = \boldsymbol{f}_{SNL}(\boldsymbol{c}(\boldsymbol{f}^*(\boldsymbol{y}), \boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*(\boldsymbol{y}))), \boldsymbol{y}), \tag{13}$$

and, synthetically, as:

$$\boldsymbol{f}^* = \boldsymbol{f}^*_{asym}(\boldsymbol{y}),$$

where $\boldsymbol{f}^*_{asym}(\boldsymbol{y})$ represents the asymmetric equilibrium link flow vector.

Hence, the optimisation model can be simplified as follows:

$$\boldsymbol{y}^\wedge = \mathrm{Arg}_{\boldsymbol{y}} \min w(\boldsymbol{y}, \boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*_{asym}(\boldsymbol{y})), \boldsymbol{f}^*_{asym}(\boldsymbol{y})) \tag{14}$$

subject to:

$$\boldsymbol{y} \in Y, \tag{15}$$

where only the topological variables, $\boldsymbol{y}$, assume the role of decision variables; in this formulation both the signal settings, $\boldsymbol{g}$, and the equilibrium traffic flows, $\boldsymbol{f}^*$, are descriptive variables.

Cascetta et al. (2006) found that the differences between solutions obtained by the local approach and solutions of the signal setting problem are not considerable, while the computation times are reduced up to 10 times.

Since the algorithms for solving the fixed-point problem (13) are studied in Cascetta et al. (1999, 2006) and they are able to jointly calculate signal settings and equilibrium flows in acceptable computation times, this paper will focus on the algorithm proposed for optimising the topological variables that can also be applied to topological network design problems, suitably modifying the decision variables and constraints.

2.2. Objective function

The objective function to minimise is the total travel time for users, calculated as:

$$w(\boldsymbol{y}, \boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*_{asym}(\boldsymbol{y})), \boldsymbol{f}^*_{asym}(\boldsymbol{y})) = \boldsymbol{c}[\boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*_{asym}(\boldsymbol{y})), \boldsymbol{f}^*_{asym}(\boldsymbol{y})]^T \boldsymbol{f}^*_{asym}(\boldsymbol{y}),$$

where each component, $c_l(\cdot)$, of the cost function vector, $\boldsymbol{c}(\cdot)$, represents the total travel time on link l (sum of running time and waiting time). Other objective functions can be adopted without problems for the functionality of the proposed approach.

2.3. Decision variables

The decision variable vector, $\boldsymbol{y}$, has as many components, $y_{(i,\,j)}$, as the non-oriented links (i,j) of the network. Conventionally, a non-oriented link is indicated with node codes (i,j), with i greater than j in terms of mere node numeration; the direction is identified by the variable value as follows:

$y_{(i,\,j)} = 0$ represents a two-way road and two oriented links $[i, j]$ and $[j, i]$ exist; to each link half the road width is assigned (for calculating link performance);

$y_{(i,\,j)} = 1$ represents a one-way road, with the direction from node i to node j; hence the oriented link $[i,j]$ exists while the opposite one does not, and the whole road width is assigned to the link $[i,j]$;

$y_{(i,\,j)} = -1$ represents a one-way road, with the direction from node j to node i; hence the oriented link $[j,i]$ exists while the opposite one does not, and the whole road width is assigned to the link $[j,i]$.

For one-lane roads, the variable $y_{(i,\,j)}$ can assume only the values 1 or -1 and this information is considered in the constraints. An example for part of an urban network is reported in Fig. 1.

In this model, we do not consider the possibility of assigning different widths (or lane number) to different directions (e.g. one lane to one direction and two lanes to the other); this is made possible by increasing the number of variable values for each link, without modifying the general model framework or the solution approach.

2.4. Constraints

The constraints of NDPs are generally classified in:

- *technical*, that avoid decision variables assuming values incongruent with an acceptable configuration of the transportation supply; these constraints always have to be considered in the problem;
- *external*, that take into account limits imposed on some results of the design (e.g. pollutant emissions, noise levels, etc.) or on the available budget; such constraints may or may not be considered in the problem depending on the design objectives and the kinds of interventions on the network;
- *assignment*, that link the descriptive variables (traffic flows) to decision ones; an assignment constraint is always present in transportation network design problems.

In the proposed model external constraints are not considered since the interventions on the network are only operational and their costs can be really neglected. Moreover, limits on the external impacts of road transportation are not considered, but their insertion in the problem does not modify the general formulation and solution algorithm.

The following technical constraints are, instead, considered in the problem:

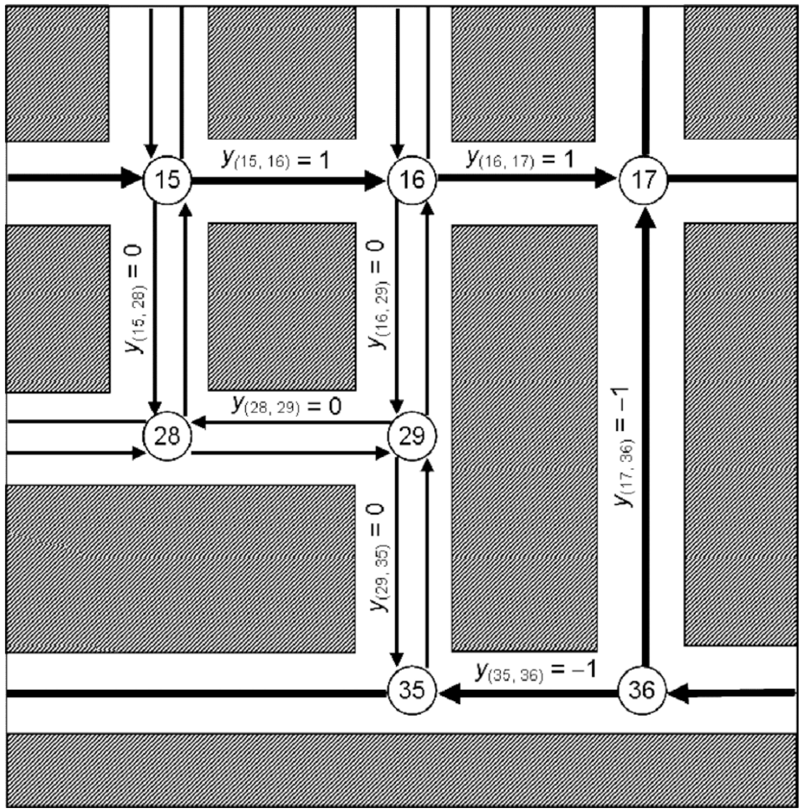

Fig. 1. Example of network layout.

(i) *connectivity constraint*, under which at least one path links each origin–destination pair; this is required to avoid the algorithm leading to a disconnected network;
(ii) *network constraints*, under which each node has at least one entering link and one exiting link; this is required to avoid other unacceptable solutions.

In the proposed model, the assignment constraint is given by Eq. (13) which jointly calculates equilibrium traffic flows and signal settings:

$$\boldsymbol{f}^* = \boldsymbol{f}_{SNL}(\boldsymbol{c}(\boldsymbol{f}^*(\boldsymbol{y}), \boldsymbol{g}_{loc\,opt}(\boldsymbol{f}^*(\boldsymbol{y}))), \boldsymbol{y}). \tag{13}$$

This constraint is automatically verified inside the solution algorithm since for each examined solution we perform a stochastic asymmetric assignment procedure that gives both equilibrium traffic flows and signal settings (see Cascetta et al., 1999, 2006). By adopting this constraint, signal settings are converted from decision to descriptive variables.

The connectivity and network constraints are not expressible in closed form, but the respect of them can be verified inside the solution algorithm any time a solution is determined. Indeed, if the network is not connected, some inconsistencies on the demand assignment arise (part of the demand is not assigned to paths). The respect of network constraints is assured if all nodes have at least an entering link and an exiting link (this control is easily implementable). Both technical constraints are verified inside the solution algorithm considering infeasible the solutions generated by the procedure that do not respect them.

2.5. Descriptive variables

The descriptive variables of the problem are the asymmetric equilibrium traffic flows, $\boldsymbol{f}^*_{asym}$, and the signal settings, $\boldsymbol{g}_{loc\,opt}$, optimised by the local approach. Both variable vectors are obtained by solving the fixed-point problem (13).

In order to reduce the complexity of the problem and speed up the fixed-point problem solution, all signalised intersections are assumed to have a two-phase plan and pre-timed cycle length. Under this assumption, vector $\boldsymbol{g}$ has as many components, g_h, as the network's signalised intersections:

$$g_h = eg^a_h / EC_h = (EC_h - eg^b_h)/EC_h \quad \forall \text{ signalised intersection } h$$

with:

$$EC_h = C_h - L_h,$$

where g_h is the signal setting variable for intersection h; C_h is the cycle length for intersection h; L_h is the total lost time for intersection h; eg^a_h is the effective green in phase a for intersection h; eg^b_h is the effective green in phase b for intersection h; EC_h is the effective cycle length for intersection h.

The solution of the fixed-point problem (13) is studied in depth by Cascetta et al. (1999, 2006).

3. Solution algorithm

The algorithm proposed for solving the topological variable optimisation problem is based on the meta-heuristic technique called *Scatter Search* (see Laguna, 2002; Glover et al., 2003). This allows discrete optimisation models to be solved, overcoming the boundaries of local optimisation. The method addresses the optimal solution search to regions that are not explored by discrete local search algorithms (e.g. neighbourhood search techniques). A scatter search method for solving a general network design problem for undirected networks has been proposed by Alvarez et al. (2005).

In the following subsections some useful basic definitions will be summarised (Section 3.1), the Neighbourhood Search algorithm, used as a subroutine of the proposed algorithm, will be described (Section 3.2) and an algorithm based on scatter search technique will be proposed for solving the topological variable optimisation problem (Section 3.3).

3.1. Basic definitions

Some basic definitions are summarised below.

Definition 1. To each solution $\boldsymbol{y} \in Y$ a set of solutions $N(\boldsymbol{y}) \subset Y$ is associated, called neighbourhood of $\boldsymbol{y}$. Solution $\boldsymbol{y}$ is called the centre of the neighbourhood $N(\boldsymbol{y})$.

Definition 2. Each solution $\boldsymbol{y}' \in N(\boldsymbol{y})$, called neighbour, is obtained from solution $\boldsymbol{y}$ by an elementary operation called a move; a move changes only one value of a variable of solution $\boldsymbol{y}$, generating the next solution $\boldsymbol{y}'$. Usually it is assumed that the neighbourhoods are symmetrical, that is: if $\boldsymbol{y}' \in N(\boldsymbol{y})$ then $\boldsymbol{y} \in N(\boldsymbol{y}')$.

Definition 3. A solution $\boldsymbol{y}^{\wedge}_{loc} \in Y$ is a local optimum if the objective function value $w(\boldsymbol{y}^{\wedge}_{loc})$ is less [greater] than, or equal to, in a minimisation [maximisation] problem, objective function values corresponding to all solutions belonging to its neighbourhood:

$$w(\boldsymbol{y}^{\wedge}_{loc}) \leqslant w(\boldsymbol{y}')\forall \boldsymbol{y}' \in N(\boldsymbol{y}),$$
$$[w(\boldsymbol{y}^{\wedge}_{loc}) \geqslant w(\boldsymbol{y}')\forall \boldsymbol{y}' \in N(\boldsymbol{y})].$$

Definition 4. The distance of solution $\boldsymbol{y}''$ from solution $\boldsymbol{y}'$ is the minimum number of moves needed to transform solution $\boldsymbol{y}''$ into solution $\boldsymbol{y}'$; the distance is indicated with $D(\boldsymbol{y}'' - \boldsymbol{y}')$. For symmetrical neighbourhoods it will be:

$$D(\boldsymbol{y}'' - \boldsymbol{y}') = D(\boldsymbol{y}' - \boldsymbol{y}'') \quad \forall \boldsymbol{y}', \boldsymbol{y}'' \in Y.$$

Any solution belonging to a neighbourhood has a distance equal to 1 from the centre:

$$D(\boldsymbol{y}' - \boldsymbol{y}) = 1 \quad \forall \boldsymbol{y}' \in N(\boldsymbol{y}).$$

For solving the UNDP, conventionally we assume as positive the moves that convert the value of a variable $y_{(i,j)}$ from 0 to 1, or from 1 to -1, or from -1 to 0; the opposite moves are assumed negative (see Fig. 2).

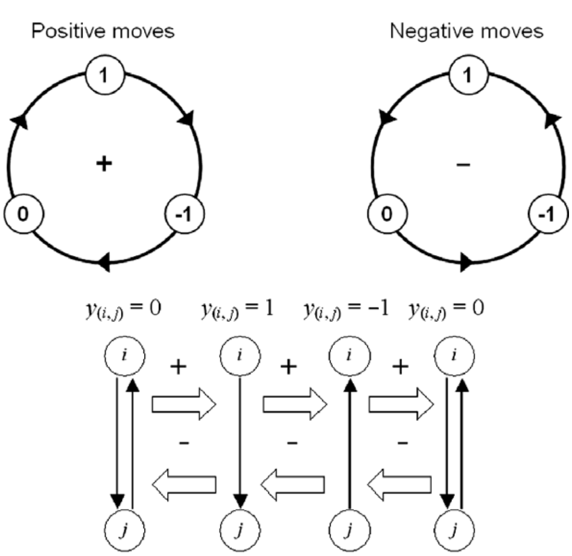

Fig. 2. 'Positive move' and 'negative move': conventional definitions.

3.2. Neighbourhood search

If $\boldsymbol{y}^k$ is a solution, the Neighbourhood Search generates the following solution $\boldsymbol{y}^{k+1}$ such that:

$$\boldsymbol{y}^{k+1} \in N(\boldsymbol{y}^k),$$

and $\boldsymbol{y}^{k+1}$ respects a specified rule.

One of the most commonly adopted rules for generating the following solution is the *steepest descent method*; it examines all neighbours, calculating their objective function values, and chooses the following solution as the one with the best value:

$$w(\boldsymbol{y}^{k+1}) = \text{Min}\{w(\boldsymbol{y}); \forall \boldsymbol{y} \in N(\boldsymbol{y}^k)\}.$$

The procedure then generates at each iteration a solution better than the previous one, choosing, among all solutions belonging to the neighbourhood, the one with the best objective function value. The procedure ends when solution $\boldsymbol{y}^k$ is a local optimum, that is when:

$$w(\boldsymbol{y}^k) \leqslant w(\boldsymbol{y}) \quad \forall \boldsymbol{y} \in N(\boldsymbol{y}^k).$$

This method is not suitable for our problem if the network has real dimensions. Indeed, in this case the variables are numerous and the neighbourhoods are very wide; evaluating at each step the objective function for all neighbours is not compatible with acceptable computation times, since each objective function evaluation requires an assignment (13).

In this paper, in order to reduce the computation times, we propose to use a random method for generating the following solution, that we call the *random descent method*. This method randomly extracts a solution from the neighbourhood and evaluates its objective function; if the new solution is better than the current one, it becomes the current solution; otherwise, another neighbourhood solution is randomly extracted and so on, until a local optimum is found. If no neighbourhood solutions improve the objective function, the last solution is a local optimum. In Section 3, the advantages of adopting the random descent method rather than the steepest descent method are numerically proved by tests on a real dimension network.

The random extraction of solution is performed by randomly extracting a variable, $y_{(i,j)}$, and randomly changing its value by a positive or negative move. In Fig. 3, we report an example of a random extraction of the following solution.

3.3. Scatter Search

Scatter Search is a *meta-heuristic* technique for solving complex combinatorial optimisation problems. Scatter Search can be adapted in several ways to several kinds of optimisation problems by suitably defining the criteria used in the *phases* of the solution procedure. A phase of Scatter Search is a mathematical or algorithmic subroutine, that operates on a solution subset generating another solution subset. Each phase (or also the sequence of phases) can be defined in different ways depending on the specific problem. In the following, the phases of the Scatter Search are examined and adapted to the UNDP.

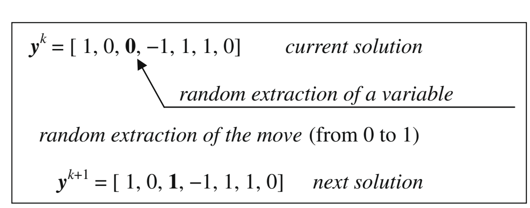

Fig. 3. Example of random extraction of a neighbour solution.

Only for better describing the phases of Scatter Search do we refer as an example to the small network reported in Fig. 4 with the decisional variables in the starting configuration.

Phase 1 – Starting set generation

In this phase, a set of solutions is generated which should have a high level of *diversity* so as to cover different parts of the solution set. The subroutine that allows us to obtain the starting set is also called the *Diversification Generation Method*, that can be generally different for each specific problem. This routine is applied in our problem as follows:

- we define a mother solution as the initial layout of the transportation network;
- from this solution, other solutions, at fixed a priori distances from the mother solution, are randomly generated; they are called base solutions;
- infeasible solutions are eliminated and substituted with other solutions (randomly generated) at the same distance from the mother one;
- the mother solution and the base solutions constitute the starting set.

		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
$\mathbf{y}^0$	= [	0,	0,	0,	0,	0,	0,	]	75

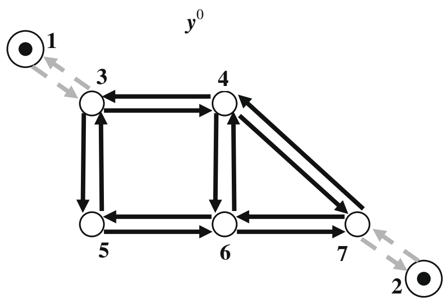

Fig. 4. Example of a network with decisional variables in the starting configuration.

An example of starting set generation for the little network of Fig. 4 is reported in Fig. 5.

For real dimension networks, we propose to generate base solutions in the following way: the maximum distance, D_{max}, between the mother solution and any other solution, which is equal to the decision variable number, is divided by an integer number, n_D; base solutions are randomly generated at distances:

$$\delta_n = \mathrm{Int}(D_{max}/n_D) \cdot n \quad (\text{for } n = 1, 2, \ldots, n_D).$$

The random generation is performed by extracting the n_D variables to change and, for each of them, extracting the move to change it.

It is possible to extract at any distance more than one solution, incrementing the base solutions; if n_s stands for the number of solutions to extract at each distance, the method can generate up to $n_D \cdot n_s$ base solutions.

Phase 2 – Improvement in current solutions

In this phase, from any current solution an improved solution is generated by an algorithmic subroutine that is also called the *improvement method*.

Several improvement methods can be adopted; in this paper we adopt and compare two neighbourhood search methods: the *steepest descent method* (or Hill Climbing) and the *random descent meth-*

			$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
Mother solution	$\mathbf{y}^0$	= [	0,	0,	0,	0,	0,	0,	]	85
Base solution dist. 1	$\mathbf{y}^{d1}$	= [	0,	**1**,	0,	0,	0,	0,	]	75
Base solution dist. 2	$\mathbf{y}^{d2}$	= [	**1**,	0,	0,	**–1**,	0,	0,	]	90
Base solution dist. 3	$\mathbf{y}^{d3}$	= [	**1**,	0,	**–1**,	**1**,	0,	0,	]	60
Base solution dist. 4	$\mathbf{y}^{d4}$	= [	**1**,	0,	**–1**,	**1**,	0,	**–1**,	]	80

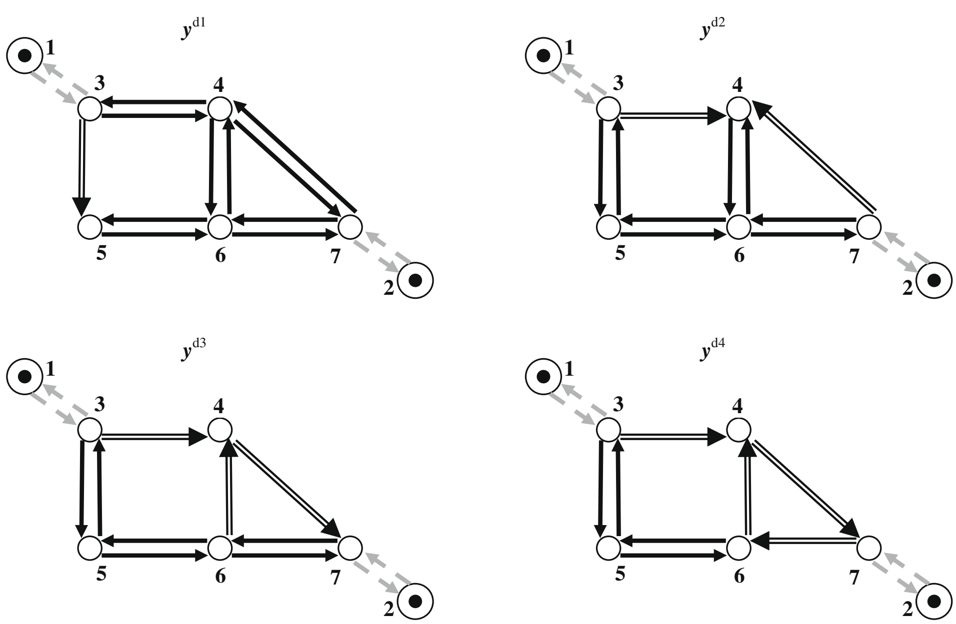

Fig. 5. Example of starting set generation.

od (see Section 3.2). The improved solutions generated by both methods are local optima.

The random descent method should lead to a local optimum in less time with respect to the steepest descent method (the two optima should be generally different), at least for real dimension networks, since the descent direction should be generated by examining fewer solutions belonging to the neighbourhood. Vice versa, the steepest descent method should lead to a local optimum in fewer descent steps but each time it has to examine all solutions of a neighbourhood and hence the total time should be higher. Indeed, examination of a solution requires a time consuming asymmetric assignment procedure.

In Fig. 6, only from a formal point of view both methods are graphically compared.

Phase 3 – Reference set generation or updating

A *reference set* is generated by selecting all improved solutions (local optima) generated in the previous phase or, if they are too numerous, only part of them; in this second case, the selection should take account of objective function values (*good solutions*) and diversity (*scattered solutions*). Especially for real networks, it is preferable to limit the dimension of the reference set fixing the maximum number of its solution. In this case, it is possible to fix a maximum number of good solutions and a maximum number of scattered solutions; the reference set will be built up by good solutions with better values of objective function, and by scattered solutions with maximum distances from the best solution, both until the maximum number of solutions is reached.

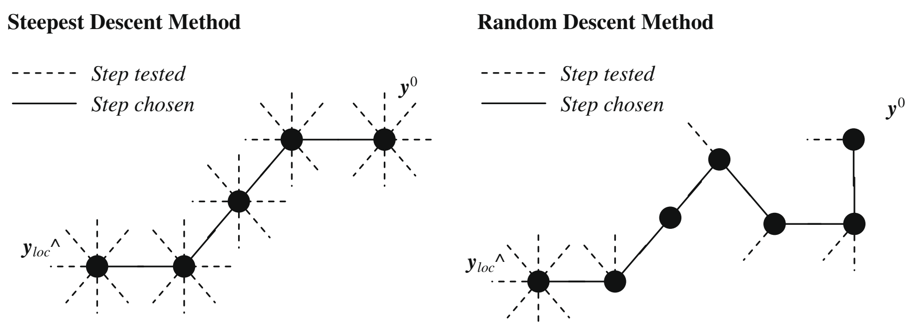

Fig. 6. Graphically comparison between random and steepest descent methods.

Solution					$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		*w*(.)	*Dist.*
1	*Loc. opt. 1*	y_{loc}^1	=	[	0,	1,	0,	0,	1,	1,	]	35	0
2	*Loc. opt. 2*	y_{loc}^2	=	[	–1,	1,	0,	–1,	1,	1,	]	40	2
3	*Loc. opt. 3*	y_{loc}^3	=	[	–1,	1,	–1,	0,	1,	1,	]	50	2
4	*Loc. opt. 4*	y_{loc}^4	=	[	–1,	0,	0,	–1,	0,	0,	]	70	5

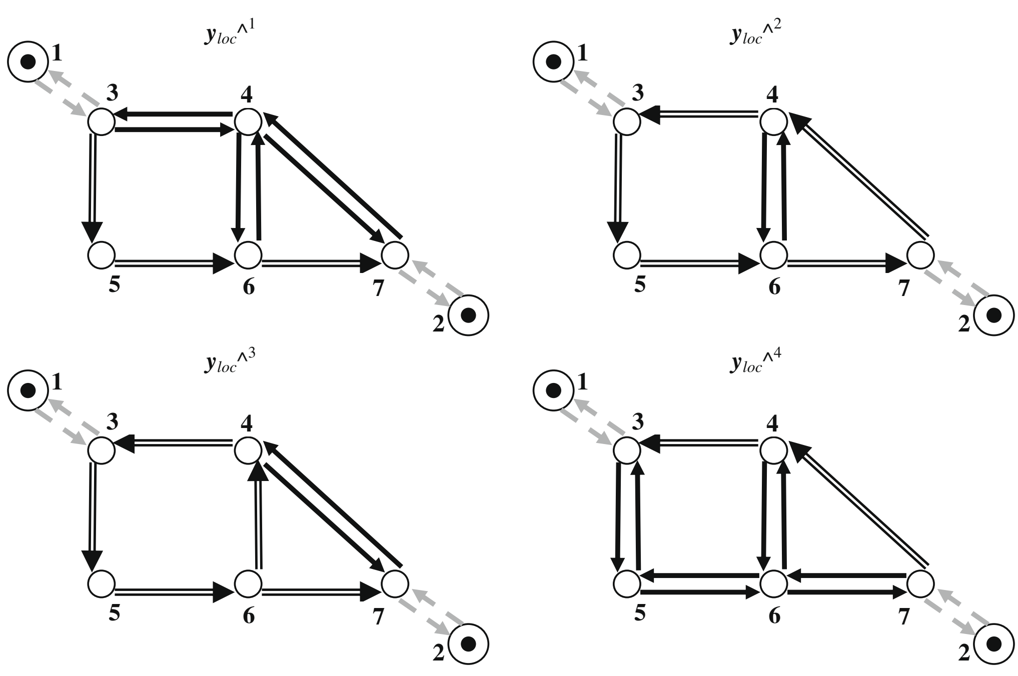

Fig. 7. Example of reference set generation.

Operating in this way, the reference set will comprise good solutions, as regards the objective function value, and scattered solutions that allow the search to be extended to regions that cannot otherwise be explored. The subroutine that generates, at the first iteration, or updates, in subsequent iterations, the reference set is called the *Reference Set Update Method*.

An example of reference set generation for the small network in Fig. 4 is reported in Fig. 7.

SUBSET 1

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
1	$\boldsymbol{y}_{loc}$^1 = [	0,	1,	0,	0,	1,	1,	]	35
2	$\boldsymbol{y}_{loc}$^2 = [	–1,	1,	0,	–1,	1,	1,	]	40

SUBSET 2

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
1	$\boldsymbol{y}_{loc}$^1 = [	0,	1,	0,	0,	1,	1,	]	35
3	$\boldsymbol{y}_{loc}$^3 = [	–1,	1,	–1,	0,	1,	1,	]	50

SUBSET 3

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
1	$\boldsymbol{y}_{loc}$^1 = [	0,	1,	0,	0,	1,	1,	]	35
4	$\boldsymbol{y}_{loc}$^4 = [	–1,	0,	0,	–1,	0,	0,	]	70

SUBSET 4

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
2	$\boldsymbol{y}_{loc}$^2 = [	–1,	1,	0,	–1,	1,	1,	]	40
3	$\boldsymbol{y}_{loc}$^3 = [	–1,	1,	–1,	0,	1,	1,	]	50

SUBSET 5

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
2	$\boldsymbol{y}_{loc}$^2 = [	–1,	1,	0,	–1,	1,	1,	]	40
4	$\boldsymbol{y}_{loc}$^4 = [	–1,	0,	0,	–1,	0,	0,	]	70

SUBSET 6

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
3	$\boldsymbol{y}_{loc}$^3 = [	–1,	1,	–1,	0,	1,	1,	]	50
4	$\boldsymbol{y}_{loc}$^4 = [	–1,	0,	0,	–1,	0,	0,	]	70

SUBSET 7

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
1	$\boldsymbol{y}_{loc}$^1 = [	0,	1,	0,	0,	1,	1,	]	35
2	$\boldsymbol{y}_{loc}$^2 = [	–1,	1,	0,	–1,	1,	1,	]	40
3	$\boldsymbol{y}_{loc}$^3 = [	–1,	1,	–1,	0,	1,	1,	]	50

SUBSET 8

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
1	$\boldsymbol{y}_{loc}$^1 = [	0,	1,	0,	0,	1,	1,	]	35
3	$\boldsymbol{y}_{loc}$^3 = [	–1,	1,	–1,	0,	1,	1,	]	50
4	$\boldsymbol{y}_{loc}$^4 = [	–1,	0,	0,	–1,	0,	0,	]	70

SUBSET 9

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$		$w(.)$
1	$\boldsymbol{y}_{loc}$^1 = [	0,	1,	0,	0,	1,	1,	]	35
2	$\boldsymbol{y}_{loc}$^2 = [	–1,	1,	0,	–1,	1,	1,	]	40
4	$\boldsymbol{y}_{loc}$^4 = [	–1,	0,	0,	–1,	0,	0,	]	70

Fig. 8. Example of solution subset generation.

SUBSET 8

Solution		$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$	$w(.)$	$w(.)$ *ratio*	*Solution score*
1	$\mathbf{y}_{loc}$^1 = [	0,	1,	0,	0,	1,	1,]	35	0.226	0.774
3	$\mathbf{y}_{loc}$^3 = [	−1,	1,	−1,	0,	1,	1,]	50	0.323	0.677
4	$\mathbf{y}_{loc}$^4 = [	−1,	0,	0,	−1,	0,	0,]	70	0.451	0.549
							Sum	155		

Variable value score	$y_{(3,4)}$	$y_{(3,5)}$	$y_{(4,6)}$	$y_{(4,7)}$	$y_{(5,6)}$	$y_{(6,7)}$
0	0.774	0.549	**1.323**	**1.451**	0.549	0.549
1	0	**1.451**	0	0	**1.451**	**1.451**
−1	**1.226**	0	0.677	0.549	0	0

Combined solution	[	−1,	1,	0,	0,	1,	1,	]

Combined solution

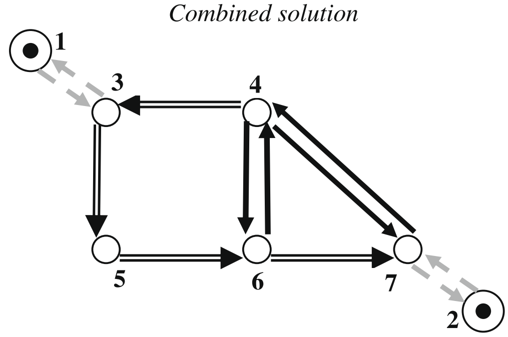

Fig. 9. Example of solution combination.

Phase 4 – Solution subset generation

In this phase, some solution subsets are generated, consisting of two or more solutions belonging to the reference set, which will be *combined* in the subsequent phase in order to generate other solutions. If n_{refset} indicates the number of solutions in the reference set, the following subsets can be generated:

(i) all subsets containing 2 solutions;
(ii) subsets containing 3 solutions obtained by adding to 2-solution subsets the best solution among those not contained in them;
(iii) the subset containing 4 solutions obtained by adding to 3-solution subsets the best solution among those not contained in them;
(iv) and so on until a subset containing $n_{ref\ set}$ solutions is generated.

As for reference set generation, if the reference set solutions are numerous, the maximum number of subsets can be fixed a priori and they can be randomly generated.

An example of solution subset generation from the reference set of Fig. 7 is reported in Fig. 8.

Phase 5 – Solution combination

In this phase, the solutions of each subset are combined. The *Solution Combination Method* may differ depending on the kind of problem and generally leads to generate one solution from each subset. The method generally associates a *score* to each value that can be assumed by a variable; this score has to take account of objective function values of solutions of the subset and of the times that the specific value is assumed by the variable. The combined solution obtained from the subset will be that in which every variable assumes the value with the best score.

In our problem the new solution is generated as follows:

(a) a *solution score* is associated to each solution of the subset as follows: the ratio between the objective function value corresponding to the solution and the sum of objective function values of all subset solutions is calculated (objective function ratio); since we are dealing with a minimisation problem the solution score is calculated as 1 less the objective function ratio;
(b) a *variable value score* is associated to each value (0, 1 or −1) that can be assumed by a variable $y_{(i,j)}$ as the sum of the relative values of solutions in which that variable assumes that specific value;
(c) the combined solution is generated such that any variable, $y_{(i,j)}$, assumes the best variable value score.

An example of solution combination from a solution subset of Fig. 8 is reported in Fig. 9.

The solutions obtained in phase 5 are improved (phase 2), generating a new reference set. The procedure ends when the reference sets in two successive iterations are equal or when a fixed a priori number of iterations is reached. All solutions belonging to the last reference set are local optima. Among them, the one with the best objective function value can be chosen.

4. Numerical results

The proposed model and algorithm were tested on the urban network of Benevento (Italy); Benevento is a town in the south of Italy with about 63,000 inhabitants. The network graph has 34 ramps and motorway links that cannot be changed, 141 urban road segments on which we can operate (they are the decision variables of our problem), 104 real nodes and 36 centroid nodes, representing trip origins and destinations (1260 OD pairs). Of the 141 road segments 12 must be only one-way roads: their variables $y_{(i,\ j)}$ can assume only 1 or -1 values. There are 17,896 peak-hour trips.

The initial layout of the network is reported in Fig. 10: the black nodes are the centroids, the white nodes are the real nodes, the dashed lines are the connectors, the double lines represent the urban motorways and the other lines are the urban roads. The picture also shows the links that in the current layout are one-way.

The travel time on each oriented link $[i,j], T_{[i,\ j]}$, is given by summing the running time, $RT_{[i,\ j]}$, and the intersection waiting time (delay), $WT_{[i,\ j]}$:

$$T_{[i,j]}(f_{[i,j]}) = RT_{[i,j]} + WT_{[i,j]}(f_{[i,j]}),$$

where $f_{[i,\ j]}$ is the flow on the oriented link $[i,j]$.

The running time, which for the sake of simplicity is assumed independent of link flows, is calculated as:

$$RT_{[i,j]} = \frac{L_{(i,j)}}{[28 + 2.5 \cdot (1 + y_{(i,j)}) \cdot (W_{(i,j)}/2)]} \cdot 3600 \quad \text{if } y_{(i,j)} = 0/1,$$

$$RT_{[i,j]} \text{ is not defined because the link does not exist} \quad \text{if } y_{(i,j)} = -1,$$

$$RT_{[j,i]} = \frac{L_{(i,j)}}{[28 + 2.5 \cdot (1 - y_{(i,j)}) \cdot (W_{(i,j)}/2)]} \cdot 3600 \quad \text{if } y_{(i,j)} = 0/-1,$$

$$RT_{[j,i]} \text{ is not defined because the link does not exist} \quad \text{if } y_{(i,j)} = 1,$$

where $RT_{[i,\ j]}$ is the running time (s) of the oriented link $[i,j]$; $L_{(i,\ j)}$ is the length (kilometer) of the non-oriented link (i,j); $W_{(i,\ j)}$ is the width (meter) of the non-oriented link (i,j).

The waiting time is calculated by the HCM (TRB, 2000) formula:

$$WT_{[i,j]}(f_{[i,j]}) = \frac{C_{[i,j]} \cdot (1 - \mu_{[i,j]})^2}{2 \cdot \left(1 - \frac{f_{[i,j]}}{s_{[i,j]}}\right)} + \frac{T}{4} \cdot \left[(X_{[i,j]} - 1) + \sqrt{(X_{[i,j]} - 1)^2 + \frac{4 \cdot X_{[i,j]}}{T \cdot \mu_{[i,j]} \cdot (s_{[i,j]}/3600)}}\right],$$

where $WT_{[i,\ j]}$ is the intersection delay (seconds/vehicle) at the final node of the oriented link $[i,j]$; $f_{[i,\ j]}$ is the flow on the oriented link $[i,j]$ in (vehicle/hours); $C_{[i,\ j]}$ is the cycle length at the final node of the oriented link $[i,j]$ (seconds); $\mu_{[i,\ j]}$ is the effective green/cycle length ratio for the oriented link $[i,j](eg_{[i,\ j]}/C_{[i,\ j]})$; $s_{[i,\ j]}$ is the saturation flow of the intersection approach $[i,j]$ (vehicle/hours); T is the oversaturation time period (seconds); $X_{[i,\ j]} = f_{[i,\ j]}/(\mu_{[i,\ j]} \cdot s_{[i,\ j]})$ is the saturation degree of the approach $[i,j]$.

The saturation flow $s_{[i,\ j]}$ is calculated by the following formulas, obtained by applying the Webster and Cobbe (1966) method in simplified fashion:

$$s_{[i,j]} = 0.9 \cdot 525 \cdot (W_{(i,j)}/2) \cdot (1 + y_{(i,j)}),$$
$$s_{[j,i]} = 0.9 \cdot 525 \cdot (W_{(i,j)}/2) \cdot (1 - y_{(i,j)}).$$

Since the waiting times, WT, are flow-dependent, also the total travel times, T, are flow-dependent and the network is congested (even though the running times, RT, are assumed constant). Therefore it is necessary to use equilibrium assignment procedures for estimating traffic flows.

The initial layout of the network is adopted as the mother solution. In our test, we assumed $n_D = 10$ (and $n_s = 1$); the distances of

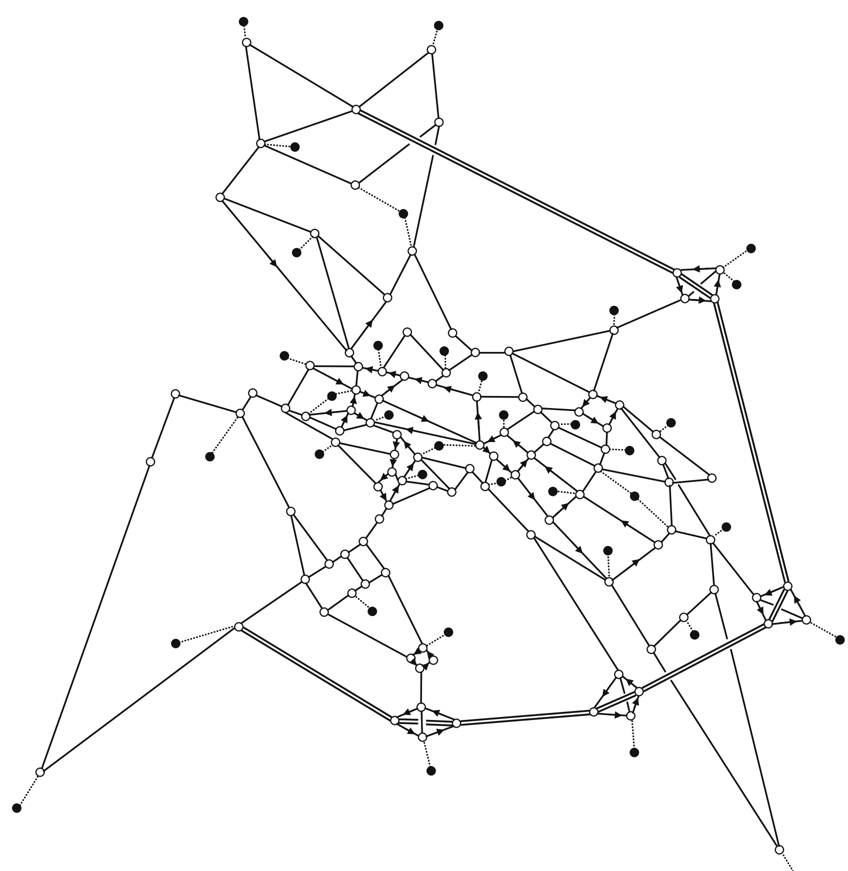

Fig. 10. Initial layout of the test network.

base solutions from the mother one are equal to 14, 28, ..., 140. Among the 10 solutions thus generated, 4 were infeasible and (after 5000 trials) a feasible solution at that distance was impossible to find. The seven solutions (the mother one and the 6 surviving solutions) are improved by the neighbourhood search method based on random extraction of the following solution, as explained in Section 2.2; Fig. 11 shows how the objective function varies as a solution is improved with the proposed neighbourhood search method.

After the first improvement phase, seven improved solutions are generated: they constitute the reference set. All the reference set's solutions are combined in the following ways:

- all solutions;
- 5 groups made by 3 solutions (randomly extracted);
- 5 groups made by 4 solutions (randomly extracted).

Obviously, there may be different ways of combining solutions, but they should depend on the number of solutions belonging to the reference set. We thus generated 11 solutions of which only 4 are feasible and are improved.

After this first scatter search iteration we obtained 12 local optima: one local optimum is obtained by improving the mother solution, 7 local optima are obtained by improving the base solutions and 4 local optima are obtained by improving the feasible combined solutions.

At the end of this first scatter search iteration, the best solution reduces the objective function by 13.82% compared with the initial layout: from 2009.4 hours to 1731.7 hours. The procedure examined 26,851 solutions; a total of 58 hours was spent on a Pentium IV computer. An ordered sequential graphic of the objective function values corresponding to all examined solutions is reported in Fig. 12.

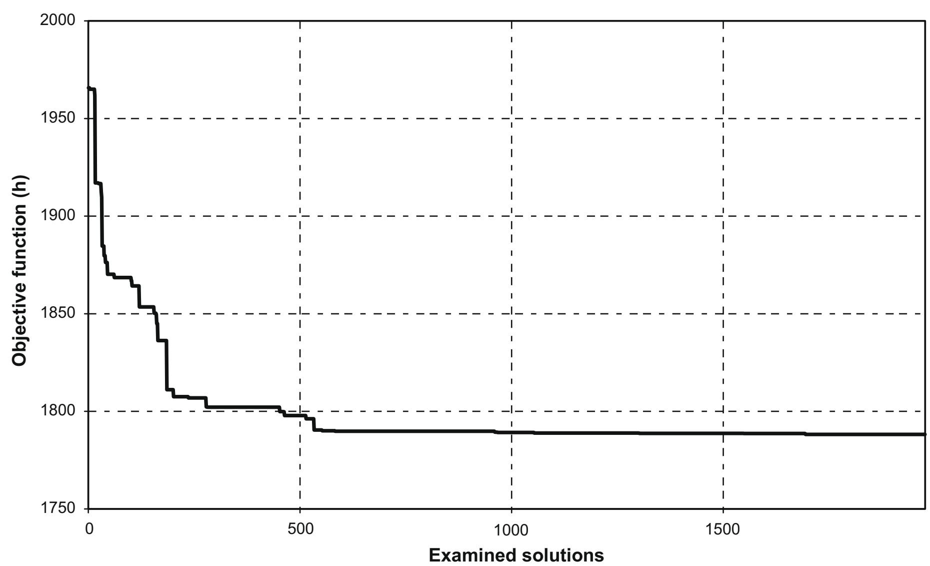

Fig. 11. Objective function reduction with a neighbourhood search phase.

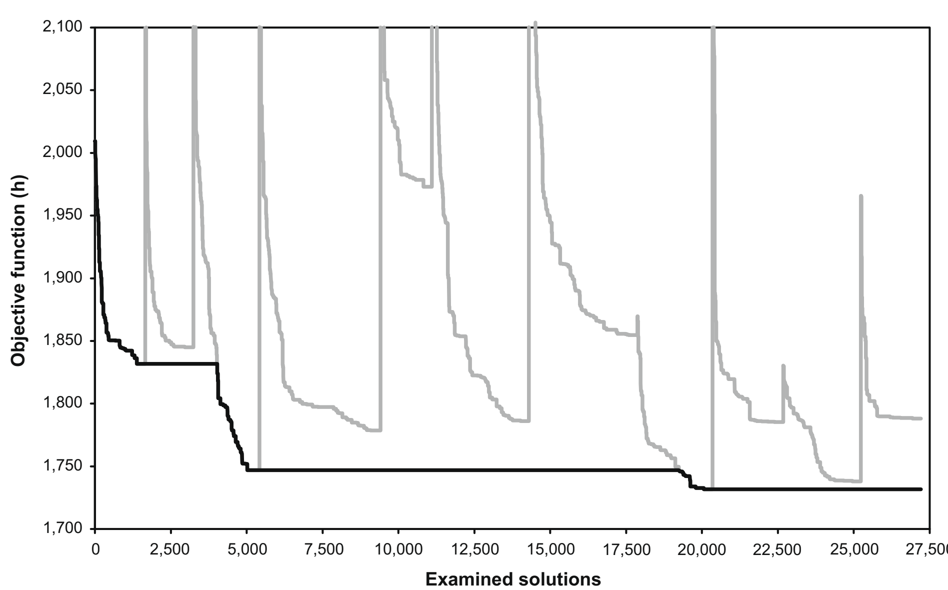

Fig. 12. Objective function reduction at first scatter search iteration.

The first scatter search iteration was performed by also adopting the steepest descent method for the neighbourhood search implementation. This test is useful for exploring differences between the two approaches and the benefits, in terms of computation times, of adopting the random generation of the next solution.

Using the steepest descent method, the procedure examined 171,751 solutions, spending 370 hours; the best solution improves the objective function over the previous case by only about 0.13%: from 1731.7 hours to 1729.4 hours. In Fig. 13, a graphical comparison of results obtained with both neighbourhood search approaches is reported.

These results show that the random method considerably reduces computation times without significantly changing the value of the objective function. The scatter search procedure continues

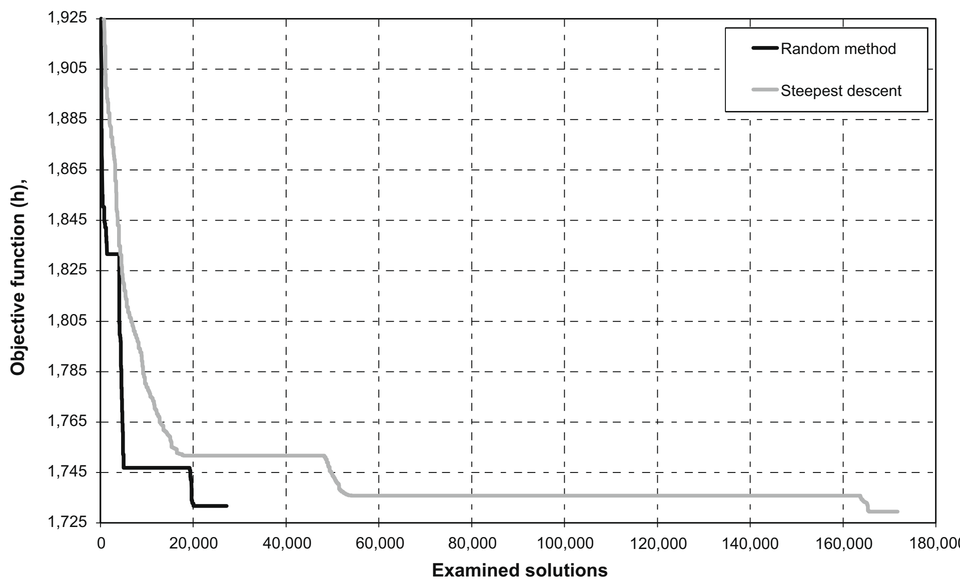

Fig. 13. Comparison of results obtained with both neighbourhood search approaches.

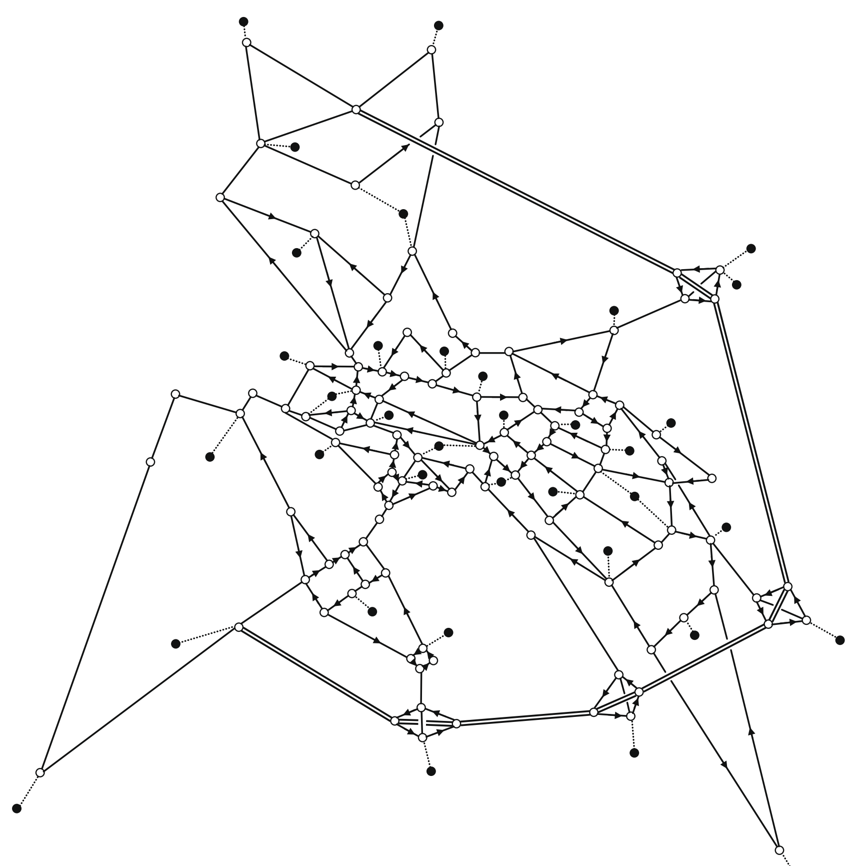

Fig. 14. Final layout of the test network.

Table 1
Algorithm performances.

	Objective function [hours]	One-way links (design variables)	Two-way links (design variables)	Mean travel time [minimum]	Speed [kilometer per hours]	Vehicle-kilometers	Saturation degree
Initial solution	2009.4	53	88	6.91	52.36	63.326	0.21
Final solution	1707.1	111	30	5.87	56.66	68.248	0.23
Variation [%]	−15.05	+109.43	−65.91	−15.05	+8.21	+7.77	+9.70

with another iteration using as the mother solution the best local optimum obtained in the previous one, and so on. After the second scatter search iteration the reduction in objective function was less than 1.45% and the search was stopped.

The proposed method examined 52,735 solutions (27,212 in the first step and 25,523 in the second), spending 113.73 hours on processing. Exhaustive research will be required to examine $1.45 \cdot 10^{65}$ solutions, spending more than $3.57 \cdot 10^{58}$ years on elaboration.

The final solution differs from the current one in 92 decision variable values (65.2% of total variables). The percentage of one-way roads rose from 37.6% to 78.7%. In Fig. 14 the final solution is reported. In Table 1, besides the objective function values, other features of the starting solution and optimal solution are compared. Note that the mean speed on the network improved by 8.21% (increasing from 52.36 km/h in the initial solution to 56.66 km/h in the final solution), total vehicle-kilometers increased by 7.77% (increasing from 63,326 to 68,248), and finally the mean saturation degree increased by 9.70% (from 0.21 to 0.23). The last two results are explained by the increase in one-way links (it increases the vehicle-kilometers on the network) and the low network congestion (0.21). In other words, minus differences in network congestion do not significantly change travel times.

These results show the improvement in urban network performance on adopting the solution obtained by applying the proposed method.

5. Conclusions

This paper proposes a Scatter Search algorithm for solving the Urban network design problem. Scatter Search is a meta-heuristic algorithm that is able to explore different regions of the feasible solution set overcoming the boundaries of local optimisation.

In the literature other meta-heuristic approaches were proposed for solving the same problem (Cantarella et al., 2006) while the Scatter Search was proposed only for solving a general bidirectional network design problem not applicable to the UNDP (Alvarez et al., 2005). Our bi-level model, which simplifies the problem and reduces its complexity, is similar to that proposed by Cantarella et al. (2006): it proposes a stochastic, rather than deterministic, approach to the asymmetric assignment problem and formally includes the asymmetric assignment constraint inside the objective function calculation, converting the problem from bi-level into single-level.

The proposed algorithm, that solves the (single-level) topological network design problem, is based on the general framework of a Scatter Search procedure but significantly reduces the computation times, applying a random descent method for the neighbourhood search adopted to improve solutions. It is thus possible to reduce the objective function evaluations that require high computation times, since they need to solve the asymmetric equilibrium problem at each time.

Our algorithm is tested on a real dimension network. The results show that it is able to find several local optima in an acceptable computation time; there is a considerable improvement in the objective function value compared to the value of the initial network layout. The random descent method significantly reduces computation times with negligible difference in the goodness of the results with respect to the steepest descent method.

Further research will be addressed to test the proposed method by varying the number of starting set solutions, in order to search for the best trade-off between computation times and goodness of solution and to compare the proposed method with other meta-heuristic approaches proposed in the literature.

References

Abdulaal, M., Le Blanc, L.J., 1979. Continuous equilibrium network design models. Transportation Research Part B 13, 19–32.

Alvarez, A.M., Gonzalez-Velarde, J.L., De-Alba, K., 2005. Scatter Search for network design problem. Annals of Operations Research 138, 159–178.

Billheimer, J.W., Gray, P., 1973. Network design with fixed and variable cost elements. Transportation Science 7, 49–74.

Boyce, E.D., Janson, B.N., 1980. A discrete transportation network design problem with combined trip distribution and assignment. Transportation Research Part B 14, 147–154.

Cantarella, G.E., Sforza, A., 1995. Network design models and methods for urban traffic management. In: Gartner, N.H., Improta, G. (Eds.), Urban Traffic Networks: Dynamic Flow Modeling and Control. Springer-Verlag, Berlin, Germany, pp. 123–153.

Cantarella, G.E., Vitetta, A., 2006. The multi-criteria road network design problem in an urban area. Transportation 33, 567–588.

Cantarella, G.E., Improta, G., Sforza, A., 1991. Road network signal setting: Equilibrium conditions. In: Papageorgiou, M. (Ed.), Concise Encyclopedia of Traffic and Transportation Systems. Pergamon Press, Oxford, United Kingdom, pp. 366–371.

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, 1682–1695.

Cascetta, E., 2001. Transportation Systems Engineering: Theory and Methods. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Cascetta, E., Gallo, M., Montella, B., 1999. An asymmetric SUE model for the combined assignment-control problem. In: Meersman, H., van de Voorde, E., Winkelmans, W. (Eds.), Selected Proceedings of 8th WCTR, vol. 2. Pergamon Press, Amsterdam, The Netherlands, pp. 189–202.

Cascetta, E., Gallo, M., Montella, B., 2006. Models and algorithms for the optimization of signal settings on urban networks with stochastic assignment. Annals of Operations Research 144, 301–328.

Chen, M., Alfa, A.S., 1991. A network design algorithm using a stochastic incremental traffic assignment approach. Transportation Science 25, 215–224.

Chiou, S.W., 1999. Optimization of area traffic control for equilibrium network flows. Transportation Science 33, 279–289.

Chiou, S.W., 2005. Bilevel programming for the continuous transport network design problem. Transportation Research Part B 39, 361–383.

Cho, H.J., Lo, S.C., 1999. Solving bilevel network design problem using linear reaction function without nondegeneracy assumption. Transportation Research Record 1667, 96–106.

Cruz, F.R.B., MacGregor Smith, J., Mateus, G.R., 1999. Algorithms for a multi-level network optimization problem. European Journal of Operational Research 118, 164–180.

D'Acierno, L., Gallo, M., Montella, B., 2006. Optimisation models for the urban parking pricing problem. Transport Policy 13, 34–48.

Dantzig, G.B., Maier, S.F., Harvey, R.P., Lansdowne, Z.F., Robinson, D.W., 1979. Formulating and solving the network design problem by decomposition. Transportation Research Part B 13, 5–17.

Davis, A.G., 1994. Exact local solution of the continuous network design problem via stochastic user equilibrium assignment. Transportation Research Part B 28, 61–75.

Drezner, Z., Wesolowsky, G.O., 2003. Network design: Selection and design of links and facility location. Transportation Research Part A 37, 241–256.

Feremans, C., Labbé, M., Laporte, G., 2003. Generalized network design problems. European Journal of Operational Research 148, 1–13.

Foulds, R.L., 1981. A multicommodity flow network design problem. Transportation Research Part B 15, 273–283.

Friesz, T.L., Cho, H.J., Mehta, N.J., Tobin, R.L., Anandalingam, G., 1992. A simulated annealing approach to the network design problem with variational inequality constraints. Transportation Science 26, 18–26.

Gao, Z., Wu, J., Sun, H., 2005. Solution algorithm for the bi-level discrete network design problem. Transportation Research Part B 39, 479–495.

Glover, F., Laguna, M., Martì, R., 2003. Scatter search. In: Ghosh, A., Tsutsui, S. (Eds.), Advances in Evolutionary Computation: Theory and Applications. Springer-Verlag, New York (NY), USA, pp. 519–537.
Harker, T.P., Friesz, T.L., 1984. Bounding the solution of the continuous equilibrium network design problem. In: Volmuller, J., Hamerslag, R. (Eds.), Proceedings of 9th International Symposium on Transportation and Traffic Theory. VNU Science Press, Utrecht, The Netherlands, pp. 233–252.
Herrmann, J.W., Ioannou, G., Minis, I., Proth, J.M., 1996. A dual ascent approach to the fixed-charge capacitated network design problem. European Journal of Operational Research 95, 476–490.
Heydecker, B.G., Khoo, T.K., 1990. The equilibrium network design problem. In: Proceedings of AIRO '90 Conference on Models and Methods for Decision Support. Sorrento (NA), Italy, pp. 587–602.
Laguna, M., 2002. Scatter search. In: Pardalos, P.M., Resende, M.G.C. (Eds.), Handbook of Applied Optimization. University Press, Oxford, United Kingdom, pp. 183–193.
Le Blanc, J.L., 1975. An algorithm for the discrete network design problem. Transportation Science 9, 183–199.
Le Blanc, J.L., Boyce, D.E., 1986. A bilevel programming algorithm for exact solution of the network design problem with user optimal flows. Transportation Research Part B 20, 259–265.
Los, M., 1979. A discrete-convex programming approach to the simultaneous optimization of land use and transportation. Transportation Research Part B 13, 33–48.
Los, M., Lardinois, C., 1982. Combinatorial programming, statistical optimization and the optimal transportation network problem. Transportation Research Part B 16, 89–124.
Magnanti, T., Wong, R., 1984. Network design and transportation planning: Models and algorithms. Transportation Science 18, 181–197.
Marcotte, P., 1983. Network optimization with continuous control parameters. Transportation Science 17, 181–197.
Meng, Q., Yang, H., 2002. Benefit distribution and equity in road network design. Transportation Research Part B 36, 19–35.
Meng, Q., Yang, H., Bell, M.G.H., 2001. An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. Transportation Research Part B 35, 83–105.
Oda, T., Otokita, T., Tsugui, T., Mashiyama, Y., 1997. Application of simulated annealing to optimization of traffic signal timings. In: Papageorgiou, M., Pouliezos, A. (Eds.), Preprints of 8th IFAC Symposium on Transportation Systems, vol. 2, Chania, Greece, pp. 733–736.
Poorzahedy, H., Abulghasemi, F., 2005. Application of ant system to network design problem. Transportation 32, 251–273.
Poorzahedy, H., Rouhani, O.M., 2007. Hybrid meta-heuristic algorithms for solving network design problem. European Journal of Operational Research 182, 578–596.
Poorzahedy, H., Turnquist, M.A., 1982. Approximate algorithms for the discrete network design problem. Transportation Research Part B 16, 45–55.
Russo, F., Vitetta, A., 2006. A topological method to choose optimal solutions after solving the multi-criteria urban network design problem. Transportation 33, 347–370.
Sheffi, Y., Powell, W.B., 1983. Optimal signal settings over transportation networks. Journal of Transportation Engineering 109, 824–839.
Solanki, S.R., Gorti, J.K., Southworth, F., 1998. The highway network design problem. Transportation Research Part B 32, 127–140.
Suwansirikul, C., Friesz, T.L., Tobin, V., 1987. Equilibrium decomposed optimization: A heuristic for the continuous equilibrium network design problem. Transportation Science 21, 254–263.
Teklu, F., Sumalee, A., Watling, D., 2007. A genetic algorithm approach for optimizing traffic control signals considering routing. Computer-Aided Civil and Infrastructure Engineering 22, 31–43.
TRB, Transportation Research Board. Highway capacity manual. National Research Council, Washington DC, USA, 2000.
Ukkusuri, S.V., Mathew, T.V., Travis Waller, S., 2007. Robust transportation network design under demand uncertainty. Computer-Aided Civil and Infrastructure Engineering 22, 6–18.
Webster, F.V., Cobbe, B.M. 1966. Traffic signals. Road Research Technical Paper No. 56. HMSO, London, United Kingdom.
Wong, C.S., 1997. Group-based optimisation of signal timings using parallel computing. Transportation Research Part C 5, 123–139.
Wong, C.S., Yang, H., 1997. Reserve capacity of a signal-controlled road network. Transportation Research Part B 31, 397–402.
Wong, C.S., Wong, W.T., Leung, C.M., Tong, C.O., 2002. Group-based optimization of a time-dependent TRANSYT traffic model for area traffic control. Transportation Research Part B 36, 291–312.
Yang, H., Yagar, S., 1995. Traffic assignment and signal control in saturated road networks. Transportation Research Part A 29, 125–139.
Ziyou, G., Yifan, S., 2002. A reserve capacity model of optimal signal control with user-equilibrium route choice. Transportation Research Part B 36, 313–323.