J Glob Optim
DOI 10.1007/s10898-015-0396-y

A modified active set algorithm for transportation discrete network design bi-level problem

Ximing Wang[1] · **Panos M. Pardalos**[1]

Received: 30 June 2015 / Accepted: 11 December 2015

Abstract Transportation discrete network design problem (DNDP) is about how to modify an existing network of roads and highways in order to improve its total system travel time, and the candidate road building or expansion plan can only be added as a whole. DNDP can be formulated into a bi-level problem with binary variables. An active set algorithm has been proposed to solve the bi-level discrete network design problem, while it made an assumption that the capacity increase and construction cost of each road are based on the number of lanes. This paper considers a more general case when the capacity increase and construction cost are specified for each candidate plan. This paper also uses numerical methods instead of solvers to solve each step, so it provides a more direct understanding and control of the algorithm and running procedure. By analyzing the differences and getting corresponding solving methods, a modified active set algorithm is proposed in the paper. In the implementation of the algorithm and the validation, we use binary numeral system and ternary numeral system to avoid too many layers of loop and save storage space. Numerical experiments show the correctness and efficiency of the proposed modified active set algorithm.

Keywords Discrete network design · Bi-level problem · Binary variable · Modified active set algorithm · Binary and ternary numeral system

1 Introduction

The transportation network design problem (NDP) is about how to modify an existing network of roads and highways in order to improve its total system travel time. For transportation network, total system travel time is sum of the multiplication of link flow and link travel

✉ Ximing Wang
x.wang@ufl.edu

Panos M. Pardalos
pardalos@ufl.edu

1 Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA

Published online: 21 December 2015

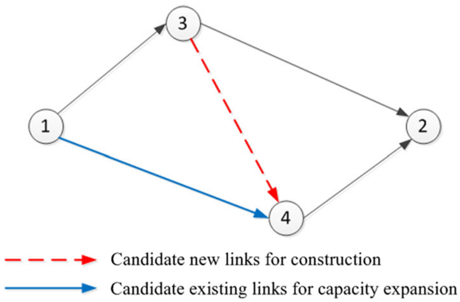

Fig. 1 Small example of discrete network design problem

time for all the links. Modifications to the network can be adding new roads, or expanding the capacities of existing roads. Studies on continuous network design problem could refer to [15]. Discrete NDP (DNDP) is a sub subject of NDP where the capacities of new roads and capacities added to existing roads are measured in candidate plans. As shown in Fig. 1, the candidate plan can be adding new links to the network (red dash line), or expanding the capacity of existing links (blue solid line). Since the construction and expansion projects are fixed, adding a fractional plan is not meaningful, the candidate road building or expansion plan can only be added as a whole.

Discrete network design problem relates to incremental network optimization, where an existing network problem is allowed to be changed within a predefined range while producing an incrementally optimal solution. Two versions of incremental minimum shortest path problem have been considered in [12], where increments are measured via arc inclusions and arc exclusions.

Discrete network design problem can be formulated into a bi-level problem with binary variables. The upper-level problem is to minimize the total system travel time with a budget constraint, which limits the number of new roads and road expansions to be added to the existing network. The lower-level problem is a User Equilibrium (UE) problem [13]. It describes how users adjust their route choices according to the new roads and road expansions.

An active set algorithm [16] has been proposed to solve the bi-level transportation discrete network design problem. It used the multipliers associated with the binary constraints to estimate the changes in system delay with different road candidate plans. Then a binary knapsack problem was constructed and solved to decide a plan to improve the system delay and also satisfy the budget constraint. This procedure kept iterated until the system delay couldn't be improved any more.

In [16], it was assumed that in the candidate plans, either new links or new lanes were added to the system. The capacity increase and construction cost of each road were based on the number of lanes, i.e., they are the same for each lane of the same link, and the maximum number of adding lanes is three. However, this might not always be the case. For example, uneven lanes have different capacities. And for different extent of road expansion, the construction costs could be different because of urban geographical change. Therefore, it is more practical to specify the capacity increase and construction cost for each candidate plan, and this is the problem analyzed in this paper. In [16], the authors use GAMS [11] to solve the UE problem and binary knapsack problem by CONOPT [6] and CPLEX [4]. In this paper, MATLAB is chosen to be the tool without using solvers. Every step is solved by numerical methods, so it is more clear and has more physical meaning to have control on each step. But this also means new ways need to be found to get some parameters that can be easily solved from GAMS but not by MATLAB. Because of the problem and tool differences, this paper proposes a modified active set algorithm, and the numerical results show the correctness and efficiency of the proposed modified active set algorithm.

Another contribution of this paper is in the implementation of the algorithm in the numerical experiments. To avoid too many layers of loop and to save storage space of parameters, a mapping between binary numeral system and decimal numeral system is used in the implementation of the modified active set algorithm. In the verification of numerical results, a similar mapping between ternary numeral system and decimal numeral system is used for the enumeration of all the possible improvement plans and calculation of their corresponding UE total system travel time. These mapping skills have the potential for solving larger problems.

The structure of this paper is as follows: Sect. 2 states the transportation discrete network design bi-level problem. Section 3 describes the original active set algorithm. Section 4 discusses the problem difference and tool difference, then proposes the modified active set algorithm. Section 5 contains the numerical experiments and the mapping details in the implementation. Section 6 concludes this paper.

2 The discrete network design bi-level problem

Different from the literature [16], we assume in this paper that each candidate plan has its own specific capacity increase and construction cost. Assuming there are two candidate plans for each new link and existing link. Let a denote the link, x_a be the link flow, t_a be the link travel time. $c_{a,0}$ denote the original capacity of link a, $\bar{A}$ denote the candidate links (including both new one and existing ones), $c_{a,k}$, $M_{a,k}$, $\forall a \in \bar{A}, k = 1, 2$ denote the capacity (or capacity increase for existing links) and construction cost for the candidate plans. $y_{a,k}, \forall a \in \bar{A}, k = 1, 2$ are binary variables, and 1 means the corresponding plan is adopted, 0 means not. For each candidate link, only one (or none) plan can be adopted, so the constraint $\sum_k y_{a,k} \leq 1$ must be hold. Another constraint is the budget with the total budget available denoted as B. The objective function of the upper-level is to minimize the total system travel time. The lower-level program is the normal UE problem under the whole improvement plan given by all the $y_{a,k}$. The DNDP bi-level problem formulation is as following:

$$\min_y \sum_a x_a^* \cdot t_a(x_a^*, c_{a,0} + c_{a,1}y_{a,1} + c_{a,2}y_{a,2}) \tag{1a}$$

$$\text{s.t.} \sum_{a \in \bar{A}} M_{a,1}y_{a,1} + M_{a,2}y_{a,2} \leq B \tag{1b}$$

$$y_{a,1} + y_{a,2} \leq 1, \ \forall a \in \bar{A} \tag{1c}$$

$$y_{a,k} \in \{0, 1\}, \ \forall a \in \bar{A}, k = 1, 2 \tag{1d}$$

$$\text{where} \tag{1e}$$

$$x_a^* = \text{argmin} \sum_a \int_0^{x_a} t_a(\varpi, c_{a,0} + c_{a,1}y_{a,1} + c_{a,2}y_{a,2}) d\varpi \tag{1f}$$

$$\text{s.t.} \sum_k f_k^{rs} = q_{rs}, \ \forall r, s \tag{1g}$$

$$f_k^{rs} \geq 0, \ \forall k, r, s \tag{1h}$$

$$x_a = \sum_{rs} \sum_k f_k^{rs} \delta_{ak}^{rs}, \ \forall a \tag{1i}$$

In model (1), the objective function (1a) is to minimize the total system travel time. Constraint (1b) is to guarantee that the total construction cost is less than or equal to the budget. Constraints (1c) and (1d) make sure that for each candidate link, only one candidate plan is adopted (when $y_{a,1} + y_{a,2} = 1$) or no action is taken (when $y_{a,1} = 0$, $y_{a,2} = 0$).

For the lower problem, (1f) to (1i) is the BMW formulation [2] of the UE problem [13]. For transportation network, user equilibrium is based on the assumption that each user wishes to minimize his/her travel time, so travel times on all used paths of each O–D pair are equal, and the travel time on any unused path is equal to or greater than the used travel time. In that case, no user can reduce his/her travel time by unilaterally changing path, so the network has become stationary, i.e. user equilibrium. The mathematical expression for UE is:

$$f_k^{rs} \cdot (c_k^{rs} - c_{\min}^{rs}) = 0, \quad \forall k, r, s \tag{2a}$$

$$c_k^{rs} - c_{\min}^{rs} \geq 0, \quad \forall k, r, s \tag{2b}$$

$$\sum_k f_k^{rs} - q_{rs} = 0, \quad \forall r, s \tag{2c}$$

$$f_k^{rs} \geq 0, \quad \forall k, r, s \tag{2d}$$

The first two formula (2a) and (2b) guarantee that the travel times on all the used paths $k(f_k^{rs} \neq 0)$ from origin r to destination s are equal to the minimum path travel time $c_{\min}^{rs}$, and the travel time on any unused path ($f_k^{rs} = 0$) is equal to or greater than the minimum path travel time, which is in accordance with the user equilibrium assumption. The third formula (2c) is the O–D flow constraint. The fourth formula (2d) is the non-negative constraints of path flow. Based on KKT conditions, the above expression (2) can be transformed into BMW formulation [2] as shown in (1f) to (1i) where x_a^* is the optimal link flow in user equilibrium.

When $y_{a,k}$ are fixed, the lower UE problem can be solved by Frank–Wolfe algorithm [3,7,9]. Frank Wolfe algorithm is an iterative first-order optimization algorithm for constrained convex optimization problems. According to BMW formulation of the UE problem, to minimize the total system travel time, the O–D flow should be assigned to the shortest path connecting that O–D pair, i.e., all-or-nothing assignment. Dijkstras Algorithm [1,5,14] can be used to find the shortest path. The step size can be determined by conducting the line search using bisection method or golden section search.

3 The original active set algorithm

Based on the assumption that capacity increase and construction cost of each road are based on the number of lanes, Zhang et.al. [16] proposed an active set approach to solve DNDP problem. The binary variables $y_{a,k}$ were divided into two active sets:

$$\Omega_0 = \{(a, k) : y_{a,k} = 0\} \tag{3a}$$

$$\Omega_1 = \{(a, k) : y_{a,k} = 1\} \tag{3b}$$

For given active sets, the improvement plan is fixed, so the lower-level UE can be solved and the total system travel time can be obtained. The basic idea of the active set approach is to move the elements between these two sets under the constraint of budget and to make the total system travel time smaller.

In the literature, the road design plan is based on the number of lanes. The construction cost on each lane of the same link is equal. The number of additional lanes is assumed to be three or less, and two binary variables $y_{a,1}$ and $y_{a,2}$ are used to determine the number of new lanes to add to link $a.y_{a,1} + 2y_{a,2}$ varies between zero and three when $y_{a,1}$ and $y_{a,2}$ are either 0 or 1. And with this notation, $y_{a,1}$ and $y_{a,2}$ can both be 1 (three new lanes). So the constraint $\sum_k y_{a,k} \leq 1$ does not exist for the literature problem.

Besides, the construction cost and capacity increase of adding one lane on link a is assumed to be equal and denoted by π_a and c_a^1. So the budget constraint becomes:

$$\sum_{a\in\bar{A}} \pi_a(y_{a,1} + 2y_{a,2}) \leq B \tag{4}$$

And the travel time BPR function [13] becomes:

$$t_a(x_a, y_a) = t_a^0\left\{1 + 0.15\left[\frac{x_a}{c_a^0 + (y_{a,1} + 2y_{a,2})c_a^1}\right]^4\right\} \tag{5}$$

where t_a^0 is the free-flow travel time and c_a^0 is the original capacity.

With the two active sets Ω_0 and Ω_1 to express the binary variables $y_{a,k}$, the formulation for the literature problem is as following:

$$\min_y \sum_a x_a^* \cdot t_a(x_a^*, c_a^0 + (y_{a,1} + 2y_{a,2})c_a^1) \tag{6a}$$

$$\text{s.t.} \sum_{a\in\bar{A}} \pi_a(y_{a,1} + 2y_{a,2}) \leq B \tag{6b}$$

$$y_{a,k} = 0,\ \forall(a,k) \in \Omega_0 \tag{6c}$$

$$y_{a,k} = 1,\ \forall(a,k) \in \Omega_1 \tag{6d}$$

$$\text{where} \tag{6e}$$

$$x_a^* = \text{argmin} \sum_a \int_0^{x_a} t_a(\varpi, c_{a,0} + (y_{a,1} + 2y_{a,2})c_a^1)d\varpi \tag{6f}$$

$$\text{s.t.} \sum_k f_k^{rs} = q_{rs},\ \forall r, s \tag{6g}$$

$$f_k^{rs} \geq 0,\ \forall k, r, s \tag{6h}$$

$$x_a = \sum_{rs}\sum_k f_k^{rs}\delta_{ak}^{rs},\ \forall a \tag{6i}$$

The lower-level UE program can be expressed by variational inequalities [8], so the bi-level problem can be denoted by a one-level optimization problem.

Let $\lambda_{a,k}$ and $\mu_{a,k}$ denote the multipliers associated with the constraints $y_{a,k} = 0$ and $y_{a,k} = 1$ respectively. It is suggested in the literature that the values of these multipliers estimate the changes in the system delay. Thus, if $\lambda_{a,k} < 0$ for some $(a, k) \in \Omega_0$, it may be beneficial to shift (a, k) from Ω_0 to Ω_1; if $\mu_{a,k} > 0$ for some $(a, k) \in \Omega_1$, it may be beneficial to shift (a, k) from Ω_1 to Ω_0. Let $g_{a,k} = 1$ mean shifting (a, k) from Ω_0 to Ω_1 and $h_{a,k} = 1$ mean shifting (a, k) from Ω_1 to Ω_0, the change of the total system travel time is estimated approximately as:

$$\sum_{(a,k)\in\Omega_0} \lambda_{a,k}g_{a,k} - \sum_{(a,k)\in\Omega_1} \mu_{a,k}h_{a,k} \tag{7}$$

The objective function of the whole problem is to minimize total system travel time, so to minimize the above expression (and to be negative) is to find a design plan that can have minimum total system travel time. However, because the KKT multipliers are only estimates of the changes, the above expression does not necessarily mean the real change of the total system travel time. If the minimum solution of $g_{a,k}$ and $h_{a,k}$ could not decrease the total system travel time, new $g_{a,k}$ and $h_{a,k}$ solutions need to be found to have as small value for the above expression as possible. A constraint is used to denote this procedure:

$$\sum_{(a,k)\in\Omega_0} \lambda_{a,k} g_{a,k} - \sum_{(a,k)\in\Omega_1} \mu_{a,k} h_{a,k} \geq \theta \tag{8}$$

where $\theta = \varepsilon + \sum_{(a,k)\in\Omega_0} \lambda_{a,k}\hat{g}_{a,k} - \sum_{(a,k)\in\Omega_1} \mu_{a,k}\hat{h}_{a,k}$, ε is a sufficiently small positive number, $\hat{g}_{a,k}$ and $\hat{h}_{a,k}$ are the minimum solution for the last run.

As there is also a budget constraint, the resulting active sets should be budget feasible. Thus the active sets adjusting problem is as following:

$$\min_{g,h} \sum_{(a,k)\in\Omega_0} \lambda_{a,k} g_{a,k} - \sum_{(a,k)\in\Omega_1} \mu_{a,k} h_{a,k} \tag{9a}$$

$$\text{s.t.} \sum_{(a,k)\in\Omega_0} \pi_a 2^{k-1} g_{a,k} - \sum_{(a,k)\in\Omega_1} \pi_a 2^{k-1} h_{a,k} \leq B - \sum_{(a,k)\in\Omega_1} \pi_a 2^{k-1} \tag{9b}$$

$$\sum_{(a,k)\in\Omega_0} \lambda_{a,k} g_{a,k} - \sum_{(a,k)\in\Omega_1} \mu_{a,k} h_{a,k} \geq \theta \tag{9c}$$

$$g_{a,k}, h_{a,k} \in \{0, 1\},\ \forall a \in \bar{A}, k = 1, 2 \tag{9d}$$

The above is the basic idea behind the original active set algorithm. It is a two-loop algorithm as shown below. The outer loop (Steps 0–2) iterates over the active pairs, calculates the total system travel time and it should be less than its predecessor. The inner loop (Steps 2a–2c) iterates over the feasible adjustment plans to find $g_{a,k}$ and $h_{a,k}$, if possible, that leads to a decrease in total system travel time than the current active pair. If this is not possible, the algorithm stops.

The original active set algorithm is as follows:

Original Active Set Algorithm

Step 0: Set $\Omega_0 = \{(a, k) : a \in \bar{A}, k = 1, 2\}$, and $\Omega_1 = \emptyset$.

Step 1: Solve the one-level optimization problem with (Ω_0, Ω_1), and determine $\lambda_{a,k}$ and $\mu_{a,k}$ with the constraints $y_{a,k} = 0$ and $y_{a,k} = 1$. Calculate the total system travel time TT and go to Step 2.

Step 2: Set $\theta = -\infty$ and adjust the active sets by performing the following steps:

(a) Let $(\hat{g}, \hat{h})$ solve the adjustment problem (9). If the optimal objective value is zero, stop, and the active sets in step 1 is the best design plan. Otherwise, go to Step 2b.

(b) Set

i. $\hat{\varphi} = \sum_{(a,k)\in\Omega_0} \lambda_{a,k}\hat{g}_{a,k} - \sum_{(a,k)\in\Omega_1} \mu_{a,k}\hat{h}_{a,k}$

ii. $\hat{\Omega}_0 = (\Omega_0 - \{(a, k) \in \Omega_0 : \hat{g}_{a,k} = 1\}) \cup \{(a, k) \in \Omega_1 : \hat{h}_{a,k} = 1\}$

iii. $\hat{\Omega}_1 = (\Omega_1 - \{(a, k) \in \Omega_1 : \hat{h}_{a,k} = 1\}) \cup \{(a, k) \in \Omega_0 : \hat{g}_{a,k} = 1\}$

(c) Solve a UE problem with $(\hat{\Omega}_0, \hat{\Omega}_1)$. If the total system travel time is less than TT, go to Step 2d. Otherwise, set $\theta = \varepsilon + \hat{\varphi}$, and return to Step 2a.

(d) Set $\Omega_0 = \hat{\Omega}_0$, $\Omega_1 = \hat{\Omega}_1$. Go to Step 1.

---

4 The modified active set algorithm

Different from [16], we consider a more general case and assume that the road design plan is expressed by two candidate plans for each new link and existing link. For each candidate link, only one (or none) plan can be adopted, so the constraint $\sum_k y_{a,k} \leq 1$ must be hold. Besides, to provide a more direct understanding and control of the algorithm and running procedure, MATLAB is used instead of GAMS, the multipliers and the Step 2a problem are solved without using solvers. These differences lead to three problems: how to express $\sum_k y_{a,k} \leq 1$ in the algorithm, how to calculate multipliers $\lambda_{a,k}$ and $\mu_{a,k}$, and how to solve Step 2a.

4.1 Expression of one-or-none constraint

For our problem, the constraint

$$y_{a,1} + y_{a,2} \leq 1, \ \forall a \in \bar{A} \tag{10}$$

must hold for every candidate link. The active set algorithm is an iterative procedure. Assume $y_{a,k} = 0$, then according to the definition of $g_{a,k}$ and $h_{a,k}$, $g_{a,k}$ can be either 0 or 1, $h_{a,k}$ must be 0. Since $g_{a,k} = 1$ means shifting (a, k) from Ω_0 to Ω_1, the new $\hat{y}_{a,k}$ is equal to $g_{a,k}$, and since $y_{a,k} = 0$, it can also be expressed as $\hat{y}_{a,k} = y_{a,k} + g_{a,k}$. Similarly, when $y_{a,k} = 1$, then $h_{a,k}$ can be either 0 or 1, $g_{a,k}$ must be 0. $h_{a,k} = 1$ means shifting (a, k) from Ω_1 to Ω_0, the new $\hat{y}_{a,k} = y_{a,k} - h_{a,k}$.

Combining the above two situations, the evolution of $y_{a,k}$ becomes:

$$\hat{y}_{a,k} = y_{a,k} + g_{a,k} - h_{a,k} \tag{11}$$

In the iterative procedure, the constraint $\sum_k y_{a,k} \leq 1$ should be hold for every iteration to make sure the solution is practical. So the following constraint could be added to Step 2a to express $\sum_k y_{a,k} \leq 1$:

$$\sum_k y_{a,k} + g_{a,k} - h_{a,k} \leq 1, \ \forall a \in \bar{A} \tag{12}$$

4.2 Calculation of multipliers

For the meaning of the multipliers, it is indicated in [16] that the values of multipliers estimate the changes in the system delay. Larsson and Patriksson [10] indicated that the Lagrange multiplier values are the shadow prices for the constraints, that is, the sensitivities of the objective function with respect to the right-hand side of the constraints. MATLAB cannot get multiplier values automatically. However, from the above meaning of the multipliers, by changing the value of $y_{a,k}$ and calculating the UE total travel time, then the difference with the original UE travel time is an estimation of the multiplier.

Since $\lambda_{a,k}$ and $\mu_{a,k}$ denote the multipliers associated with the constraints $y_{a,k} = 0$ and $y_{a,k} = 1$, it can be expanded that when $y_{a,k} = 1$, $\lambda_{a,k} = 0$; when $y_{a,k} = 0$, $\mu_{a,k} = 0$. Using z to express the original objective function value (total system travel time), z' expressing the new one, a direct thought is: when $y_{a,k} = 0$, others remain the same, only change $y'_{a,k} = 1$, then $\lambda_{a,k} = (z' - z)/(1 - 0) = z' - z$; when $y_{a,k} = 1$, others remain the same, only change $y'_{a,k} = 0$, then $\mu_{a,k} = (z' - z)/(0 - 1) = z - z'$. So the calculation of $\lambda_{a,k}$ and $\mu_{a,k}$ can be expressed as:

$$\begin{cases} y_{a,k} = 0: & \lambda_{a,k} = z' - z \quad \mu_{a,k} = 0 \\ y_{a,k} = 1: & \lambda_{a,k} = 0 \qquad\quad \mu_{a,k} = z - z' \end{cases} \tag{13}$$

The above thought is quite straight forward, however, it is not practical because it might violate the constraint of $\sum_k y_{a,k} \leq 1$. Let $(0, 0)$ denote the case that $y_{a,1} = 0$ and $y_{a,2} = 0$, $(1, 0)$ denote that $y_{a,1} = 1$ and $y_{a,2} = 0$, $(0, 1)$ denote that $y_{a,1} = 0$ and $y_{a,2} = 1$, these are the only adoption plans for link a. In the $(0, 0)$ case, there would be no problem when changing one single 0 to 1. However, in the case of $(1, 0)$ and $(0, 1)$, when change the 0 to 1 to calculate $\lambda_{a,k}$, the new plan would become $(1, 1)$, which is not practical. For the calculation of $\mu_{a,k}$, since it is generated by changing 1 to 0, there is no such problem. So the above expression should be modified to calculate $\lambda_{a,k}$.

The problem happens at $(1, 0)$ and $(0, 1)$, so these two situations will be analyzed respectively. In the case of $(1, 0)$, we have $\lambda_{a,1} = 0$, $\mu_{a,2} = 0$, so only $\lambda_{a,2}$ and $\mu_{a,1}$ need to be calculated. There is no problem with the calculation of $\mu_{a,1}$. Using $z_{0,0}$, $z_{1,0}$, and $z_{0,1}$ to express the total system travel time under $(0, 0)$, $(1, 0)$ and $(0, 1)$, then $\mu_{a,1} = z_{1,0} - z_{0,0}$. For $\lambda_{a,2}$, $(1, 1)$ case is not practical, so it can only be changed to $(0, 1)$. In this case (change from $(1, 0)$ to $(0, 1)$), considering about the objective function in Step 2a (which is an approximation of the change in total system travel time), the function would become $z_{0,1} - z_{1,0} = -\mu_{a,1} + \lambda_{a,2}$, so $\lambda_{a,2} = z_{0,1} - z_{0,0}$. Similarly, in the case of $(0, 1)$, $\lambda_{a,1} = z_{1,0} - z_{0,0}$.

The calculation of $\lambda_{a,k}$ and $\mu_{a,k}$ can be summarized as:

$$\begin{cases} (0,0): & \lambda_{a,1} = z_{1,0} - z_{0,0} & \lambda_{a,2} = z_{0,1} - z_{0,0} \\ & \mu_{a,1} = 0 & \mu_{a,2} = 0 \\ (1,0): & \lambda_{a,1} = 0 & \lambda_{a,2} = z_{0,1} - z_{0,0} \\ & \mu_{a,1} = z_{1,0} - z_{0,0} & \mu_{a,2} = 0 \\ (0,1): & \lambda_{a,1} = z_{1,0} - z_{0,0} & \lambda_{a,2} = 0 \\ & \mu_{a,1} = 0 & \mu_{a,2} = z_{0,1} - z_{0,0} \end{cases} \tag{14}$$

4.3 Step 2 solving procedures

Step 2a is a binary integer optimization problem. $g_{a,k}$ and $h_{a,k}$ are decision variables and also binary variables. In the inner loop of Step 2 (Steps 2a–2c), Step 2a problem was solved iteratively with increasing θ to get the solution which can really reduce the total system travel time. Without using binary integer optimization solvers but only basic MATLAB, new approaches need to be used to solve the problem.

Since $g_{a,k}$ and $h_{a,k}$ are binary variables, and both the constraints and objective function are linear which does not require large amount of computation, a direct thought is to enumerate all the possible $g_{a,k}$ and $h_{a,k}$ that satisfy the constraints, and calculate the objective function value. The values are sorted in ascending order, and the pair with the minimum value (which should be less than or equal to zero) is the optimal solution.

The enumeration method has a bonus advantage for Step 2c. In this step, the UE total system travel time is calculated to test whether the solution really reduce the total system travel time. If not, θ is increased to make sure a new solution can be obtained. The enumeration method has a sorted group of $g_{a,k}$ and $h_{a,k}$, so with the same idea in Step 2c, if the pair with the smallest value cannot reduce the total system travel time, the pair with the second smallest value will be tested. This procedure continues until the pair of $g_{a,k}$ and $h_{a,k}$ which can really reduce the total system travel time is found. And if the smallest value is equal to zero, this means the total system travel time cannot be reduced according to the estimated change, then the algorithm stops.

4.4 The modified active set algorithm

Summarizing the above differences and corresponding solving procedures, the modified active set algorithm for our problem is as following:

Modified Active Set Algorithm

Step 0: Set $\Omega_0 = \{(a, k) : a \in \bar{A}, k = 1, 2\}$, and $\Omega_1 = \emptyset$.

Step 1: Get total system travel time TT and multiplier values $\lambda_{a,k}$ and $\mu_{a,k}$.

(a) Solve the UE problem with (Ω_0, Ω_1) using Frank–Wolfe Algorithm, and calculate the total system travel time TT.

(b) Change one $y_{a,k}$ at each time and re-solve the UE problem to calculate the values of $\lambda_{a,k}$ and $\mu_{a,k}$ according to (14).

(c) Go to Step 2.

Step 2: Find the new active sets $(\hat{\Omega}_0, \hat{\Omega}_1)$ which can reduce the total system travel time.

(a) Enumerate all the feasible pairs of $g_{a,k}$ and $h_{a,k}$ which satisfy the following constraints:

$$\sum_{a,k} M_{a,k} g_{a,k} - \sum_{a,k} M_{a,k} h_{a,k} + \sum_{a,k} M_{a,k} y_{a,k} \leq B$$
$$\sum_{k} y_{a,k} + g_{a,k} - h_{a,k} \leq 1, \ \ \forall a \in \bar{A}$$
$$g_{a,k}, h_{a,k} \in \{0, 1\}, \ \ \forall a \in \bar{A}, k = 1, 2$$

For each feasible pair of $g_{a,k}$ and $h_{a,k}$, calculate the following value:

$$\sum_{a,k} \lambda_{a,k} g_{a,k} - \mu_{a,k} h_{a,k}$$

Sort the values in ascending order, also the corresponding $g_{a,k}$ and $h_{a,k}$. If the smallest value ($\hat{g}_{a,k}$ and $\hat{h}_{a,k}$ is the corresponding pair) is zero, stop, and the active sets in step 1 is the best design plan. Otherwise, go to Step 2b.

(b) Set

i. $\hat{y}_{a,k} = y_{a,k} + \hat{g}_{a,k} - \hat{h}_{a,k}$

ii. $\hat{\Omega}_0 = \{(a, k) : \hat{y}_{a,k} = 0\}$

iii. $\hat{\Omega}_1 = \{(a, k) : \hat{y}_{a,k} = 1\}$

(c) Solve a UE problem with $(\hat{\Omega}_0, \hat{\Omega}_1)$. If the total system travel time is less than TT, go to Step 2d. Otherwise, set $\hat{g}_{a,k}$ and $\hat{h}_{a,k}$ with the pair corresponding to the second smallest value. If this value is zero, stop, the active sets in step 1 is the best design plan. Otherwise, return to Step 2b.

(d) Set $\Omega_0 = \hat{\Omega}_0$, $\Omega_1 = \hat{\Omega}_1$. Go to Step 1.

---

5 Numerical experiments[1]

The data for the Sioux Falls candidate improvement plans is shown in Fig. 2 and Tables 1, 2, and 3.

[1] Code provided on http://ximing.wix.com/ximing#!research/c1ac3

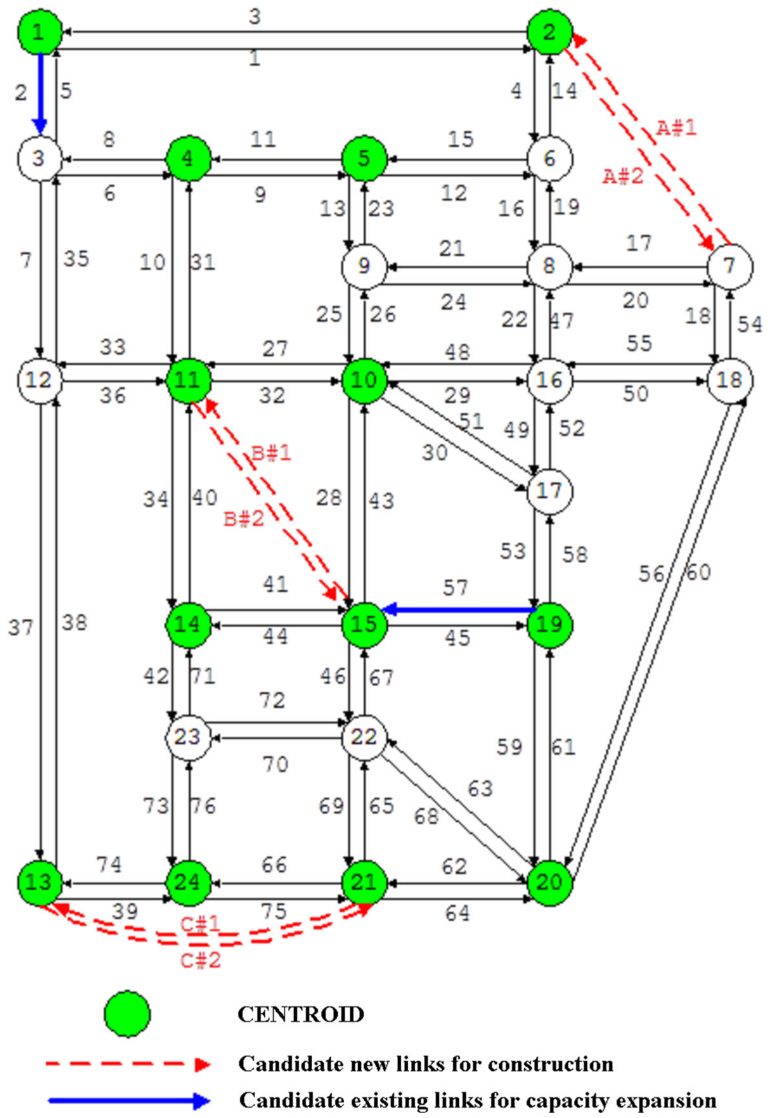

Fig. 2 Candidate highway improvement problem on the network

5.1 Solution under certain budget

When implementing our modified active set algorithm, in Step 2a, it needs an enumeration of all the feasible pairs of $g_{a,k}$ and $h_{a,k}$. Our example problem has 8 candidate links (6 new candidate links and 2 existing candidate links), and each link has 2 candidate plans. $g_{a,k}$ and $h_{a,k}$ are corresponding to $y_{a,k} = 0$ and $y_{a,k} = 1$ respectively. As $g_{a,k}$ and $h_{a,k}$ are binary variables, there are totally $2^{16} = 65{,}536$ combinations (without considering the constraints). To express this large number of enumeration is a problem. 16 layers of loop does not make sense and is too tedious. Also the algorithm needs to store $g_{a,k}$ and $h_{a,k}$ sorted by the objective function value. Binary $g_{a,k}$ and $h_{a,k}$ need very large multiple dimensional matrix to store them. In our implementation, a mapping between binary numeral system and decimal numeral system is utilized to stand for $g_{a,k}$ and $h_{a,k}$, and it is quite efficient in enumeration and storage.

Let m and n $(m + n = 16)$ denote the number of $y_{a,k} = 0$ and $y_{a,k} = 1$, then the different possibilities for $g_{a,k}$ and $h_{a,k}$ are 2^m and 2^n, respectively. The binary $g_{a,k}$ is in the form

$$(b_1, b_2, \ldots, b_m), \quad b_i \in \{0, 1\}, i = 1, 2, \ldots, m$$

The above binary value can be transformed into decimal number $2^0 b_1 + 2^1 b_2 + \cdots + 2^{m-1} b_m$. Each binary value has a unique mapping decimal number, and equally, each decimal number maps a unique binary number. So using a loop for decimal g to be from 0 to $2^m - 1$, the corresponding binary $g_{a,k}$ will cover all the possibilities of $g_{a,k}$. The decimal values will be stored in order. This will be just one vector, so it is very easy and efficient in storage. The same case is for $h_{a,k}$. A total number of $2^m \times 2^n = 2^{16} = 65{,}536$ combination will be calculated, the same as the binary case, so the enumeration is comprehensive.

When getting all the feasible pairs of $g_{a,k}$ and $h_{a,k}$ which satisfy all the constraints and their corresponding objective function value, they are sorted and as an input to Step 2b. The stored g and h are decimal numbers, in this stage, the decimal number can be transformed back into binary number $g_{a,k}$ and $h_{a,k}$ for further calculation.

When the total budget available is \$95 million. The optimal improvement plan solved by our modified active set algorithm is:

$$y_{a,k} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

The total system travel time is 1196.9 (10^3veh · min/h), less than the original network UE total system travel time 1271.3. The total construction cost is \$95 million, equal to the budget. The running time is 85.7 sec. The Step 1 outer loop has 2 iterations; the Step 2 inner loop has 1 iteration. The majority of the computational time was spent on the UE problem in Step 1.

5.2 Validation of the solutions

To check whether the solution got by our modified active set algorithm is the true optimal solution that minimizes the total system travel time, an enumeration of all the possible improvement plans and calculation of their corresponding UE total system travel time is performed. For each candidate link, there are three possibilities, so there are a total of $3^8 = 6561$ improvement plans (without considering about the budget constraint).

For this enumeration, a mapping between ternary numeral system and decimal numeral system is used. Using $p_i \in \{0, 1, 2\}, i = 1, 2, \ldots, 8$ to denote the candidate plan for each link $i. p_i = 0$ means no candidate plan is adopted, $p_i = 1$ means candidate plan 1 is adopted, $p_i = 2$ means candidate plan 2 is adopted. The improvement plan could be expressed as

$$(p_1, p_2, \ldots, p_8), \quad p_i \in \{0, 1, 2\}, i = 1, 2, \ldots, 8$$

Table 1 Original link capacity and free-flow travel time

Link number	Free-flow travel time (min)	Capacity (10^3 veh/h)	Link number	Free-flow travel time (min)	Capacity (10^3 veh/h)
1	3.60	6.02	39	2.40	10.18
2	2.40	9.01	40	2.40	9.75
3	3.60	12.02	41	3.00	10.26
4	3.00	15.92	42	2.40	9.85
5	2.40	46.81	43	3.60	27.02
6	2.40	34.22	44	3.00	10.26
7	2.40	46.81	45	2.40	9.64
8	2.40	25.82	46	2.40	20.63
9	1.20	28.25	47	3.00	10.09
10	3.60	9.04	48	3.00	10.27
11	1.20	46.85	49	1.20	10.46
12	2.40	13.86	50	1.80	39.36
13	3.00	10.52	51	4.20	9.99
14	3.00	9.92	52	1.20	10.46
15	2.40	9.90	53	1.20	9.65
16	1.20	21.62	54	1.20	46.81
17	1.80	15.68	55	1.80	39.36
18	1.20	46.81	56	2.40	8.11
19	1.20	9.80	57	2.40	4.42
20	1.80	15.68	58	1.20	9.65
21	2.00	10.10	59	2.40	10.01
22	3.00	10.09	60	2.40	8.11
23	3.00	20.00	61	2.40	6.05
24	2.00	10.10	62	3.60	10.12
25	1.80	27.83	63	3.00	10.15
26	1.80	27.83	64	3.60	10.12
27	3.00	20.00	65	1.20	10.46
28	3.60	27.02	66	1.80	9.77
29	3.00	10.27	67	2.40	20.63
30	4.20	9.99	68	3.00	10.15
31	3.60	9.82	69	1.20	10.46
32	3.00	20.00	70	2.40	10.00
33	3.60	9.82	71	2.40	9.85
34	2.40	9.75	72	2.40	10.00
35	2.40	46.81	73	1.20	10.16
36	3.60	9.82	74	2.40	11.38
37	1.80	51.80	75	1.80	9.77
38	1.80	51.80	76	1.20	10.16

Table 2 O–D trip matrix for the network (10^3 veh/h)

	1	2	4	5	10	11	13	14	15	19	20	21	22	24
1		1.32	1.32	1.32	1.08	1.10	1.25	0.99	0.95	0.90	0.59	0.59	0.77	0.74
2	1.32		1.25	1.30	1.10	1.12	0.90	0.95	0.94	1.30	0.59	0.68	0.67	0.59
4	1.32	1.25		1.32	1.08	1.07	0.95	0.90	0.84	0.80	1.62	0.64	0.59	0.80
5	1.32	1.30	1.32		1.13	0.97	0.91	0.88	0.81	0.73	0.80	0.81	0.94	0.59
10	1.08	1.10	1.08	1.13		1.33	0.90	0.99	1.32	1.17	0.95	0.90	0.97	0.59
11	1.10	1.12	1.07	0.97	1.33		0.94	1.32	1.11	0.95	0.74	0.61	1.10	1.05
13	1.25	0.90	0.95	0.91	0.90	0.94		0.87	0.86	0.68	0.59	0.62	0.67	1.32
14	0.99	0.95	0.90	0.88	0.99	1.32	0.87		1.32	1.13	0.95	0.87	0.90	1.13
15	0.95	0.94	0.84	0.81	1.32	1.11	0.86	1.32		1.32	1.27	1.14	1.32	0.91
19	0.90	1.30	0.80	0.73	1.17	0.95	0.68	1.13	1.32		1.32	1.11	1.10	0.80
20	0.59	0.59	1.62	0.80	0.95	0.74	0.59	0.98	1.27	1.32		1.32	1.32	0.61
21	0.59	0.68	0.64	0.81	0.90	0.61	0.62	0.87	1.14	1.11	1.32		1.32	1.32
22	0.77	0.67	0.59	0.94	0.97	1.10	0.67	0.90	1.32	1.10	1.32	1.32		1.13
24	0.74	0.59	0.80	0.59	0.59	1.05	1.32	1.13	0.91	0.80	0.61	1.32	1.13	

Table 3 Candidate highway plans and specifications

Link number	Free-flow travel time (min)	Capacity (10^3 veh/h)	Construction cost (million $)
Construction of new candidate links			
A#1	4.00	4.0	10
	4.00	6.0	15
A#2	4.00	4.0	10
	4.00	6.0	15
B#1	3.50	4.0	9
	3.50	6.0	14
B#2	3.50	4.0	9
	3.50	6.0	14
C#1	4.00	6.0	15
	4.00	8.0	22
C#2	4.00	6.0	15
	4.00	8.0	22
Link number	**Capacity increase (10^3 veh/h)**	**Construction cost (million $)**	
Expansion of existing candidate links			
2	2.0	3	
	4.0	6	
57	2.0	3	
	4.0	6	

Table 4 Solution under different budgets

Budget (million \$)	True optimal solution			Modified active set algorithm			
	TT (10^3 veh min/h)	Feasible enumerations	Money spent (million \$)	TT (10^3 veh min/h)	Step 1 outer iter	Step 2 inner iter	Running time (s)
114	1194.2	6561	114	1194.2	2	1	64.1
113	1194.5	6560	109	1194.5	3	2	115.1
108	1195.5	6554	107	1195.5	3	2	120.1
106	1195.7	6549	102	1195.7	3	2	131.9
101	1196.7	6508	100	1196.7	3	2	127.7
99	1196.9	6479	95	1196.9	3	2	133.3
94	1198.5	6353	90	1198.5	3	2	161.6
89	1199	6139	89	1199	3	2	157.8
88	1200.1	6075	87	1200.1	2	1	106.4
86	1200.5	5951	84	1200.5	2	1	101.9
83	1203	5719	81	1203	2	1	100.8
80	1204.1	5442	80	1204.1	3	2	141
79	1204.6	5360	79	1204.6	3	2	114.6
78	1205.7	5242	77	1205.7	3	2	138.8
76	1205.8	5033	75	1205.8	2	1	92.7
74	1206.2	4800	74	1206.2	3	2	157.4
73	1207.5	4663	72	1207.5	2	1	92.9
71	1207.9	4417	69	1207.9	2	1	89.5
68	1210.4	4010	66	1210.4	2	1	89.3
65	1211.5	3595	65	1211.5	2	1	93.9
64	1212	3469	64	1212	3	2	123.6
63	1213.1	3314	62	1213.1	2	1	95.7
61	1213.5	3048	59	1213.5	2	1	91.9
58	1215.8	2628	57	1215.8	2	1	92.2

Table 4 continued

Budget (million \$)	True optimal solution			Modified active set algorithm			
	TT (10^3 veh min/h)	Feasible enumerations	Money spent (million \$)	TT (10^3 veh min/h)	Step 1 outer iter	Step 2 inner iter	Running time (s)
56	1216	2368	56	1216	2	1	93
55	1216.2	2250	54	1216.2	2	1	88.4
53	1218.7	1993	51	1218.7	2	1	88.2
50	1219.8	1655	50	1219.8	2	1	88.6
49	1220.3	1554	49	1220.3	3	2	113.9
48	1221.4	1440	47	1221.4	2	1	88.1
46	1221.8	1257	44	1221.8	2	1	86.6
43	1224.3	989	41	1224.3	2	1	87.4
40	1227.5	773	39	1227.5	2	1	75.7
38	1230	632	36	1230	2	1	75.8
35	1233.3	477	33	1233.3	2	1	73.8
32	1235	346	30	1235	2	1	70.1
29	1237.5	238	27	1237.5	2	1	70
26	1242	161	25	1242	2	1	80.4
24	1243.9	130	24	1243.9	2	1	72.6
23	1244.5	107	22	1244.5	2	1	77.6
21	1246.9	87	19	1246.9	2	1	83.3
18	1248.7	56	16	1248.7	2	1	81.1
15	1251.2	33	13	1251.2	2	1	81.2
12	1257.4	17	10	1257.4	2	1	80.4
9	1260.7	10	9	1260.7	2	1	64.1
8	1262.5	6	6	1262.5	2	1	58.1
5	1265	3	3	1265	2	1	58.6
2	1271.3	1	0	1271.3	1	0	29.2

The above ternary number can be transformed into decimal number $3^0 p_1 + 3^1 p_2 + \cdots + 3^7 p_8$. The enumeration of the 6561 plans can use a loop for decimal number to be from 0 to 6560, the corresponding ternary number will cover all the possibilities.

In the enumeration, if the budget constraint is checked before performing UE, then there are 6377 plans that are budget feasible for \$95 million and they all need to have their UE total system travel time calculated. The result with the minimum total system travel time is the same with the plan solved by our modified active set algorithm. However, the running time is 4 h and 50 min. So our proposed modified active set approach is much more effective.

To check whether this program is effective for other budgets, firstly all the 6561 cases were calculated for their total system travel time. The total running time for the 6561 cases is 5 h and 6 min.

Then, to verify the results of our modified active set algorithm, the budget starts with the maximum total cost of all the plans, i.e. \$114 million. The resulting optimal plan also has a total money of \$114 million. Then reduce this by \$1 million, running with budget \$113 million. The resulting optimal plan has a total money of \$109 million. Then the running budget is \$108 million, and so on. The result is shown in Table 4.

The result shows a very good performance under different budgets. All the active set approach solution is the true optimal solution.

When the budget is \$95 million, the iteration count is rather small, only 2 Step1 iterations and 1 Step2 inner iteration. This means the program directly found the optimal solution by just using the combination of $\lambda_{a,k}$ and $\mu_{a,k}$. When there are 3 Step1 iterations, the direct combination solution is not the optimal solution, so more iterations were taken and achieved the optimal result. This demonstrates the correctness of the algorithm from another side.

Solving the UE problem consumes the majority of the computational time. For each budget, the number of UE calculation is: 16×(Step 1 Outer Iter) + 1× (Step 2 Inner Iter). Most of the budgets has 2 Step1 iterations and 1 Step2 iteration, so UE problem was solved 33 times. Some budgets has 3 Step1 iterations and 2 Step2 iterations, so UE problem was solved 50 times, and this is the maximum number of UE calculations for any budget for this network.

Compared with enumeration method, the budget feasible plans have thousands of counts, and each count needs to calculate their UE travel time. So for budget over \$16 million, the modified active set algorithm is more efficient than enumeration method and can get the true optimal solution within 3 min.

The budget and the corresponding minimum total system travel time are plotted in Fig. 3.

For any budget, the slope of the line connecting the left most point (budget = 0) and the budget point can represent the efficiency of money to improve the total system travel time (10^3 veh · min/h per million\$). It can been seen from Fig. 3 that the efficiency of money is gradually decreasing when the budget increases.

6 Conclusion

Transportation discrete network design problem can be formulated into a bi-level problem with binary variables and solved by active set algorithm. The original active set algorithm assumed that the capacity increase and construction cost of each road are the same for each lane of the same link. We consider a more general case in this paper when the capacity increase and construction cost are specified for each candidate plan. Every step is solved by numerical methods instead of optimization solvers to provide a more direct understanding and control

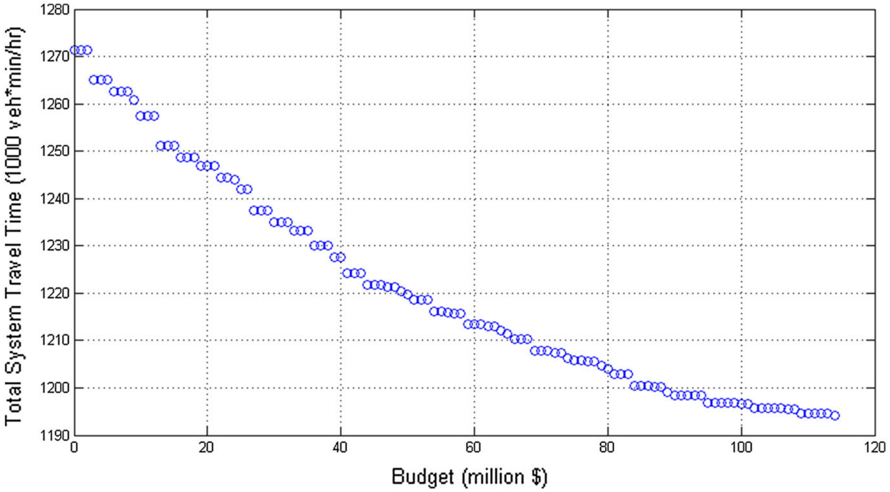

Fig. 3 Budget and the corresponding minimum total system travel time

of the algorithm and running procedure. Because of the problem and tool differences, we modify the original active set algorithm and propose our own algorithm, and also use binary and ternary numeral system in the implementation and validation to avoid too many layers of loop and save storage space. Numerical experiments show the correctness and efficiency of the proposed modified active set algorithm.

In the validation, the proposed algorithm is effective for different budgets, but it will be more convincing to have more validations on other networks to demonstrate the effectiveness of the algorithm. Because the multiplier only considers the single change of $y_{a,k}$, the combination of the change when multiple $y_{a,k}$ are changing is only a very rough estimation of the true change. That is why Step 2 needs an inner loop. In further study, more networks could be experimented to give firmer justification on the effectiveness of the proposed modified active set algorithm.

Acknowledgments We are grateful to Yafeng Yin for providing the problem and the original active set algorithm, and also the valuable discussions with him. P. M. Pardalos was partially supported by LATNA Laboratory, NRU HSE, RF government grant, ag. 11.G34.31.0057, National Research University Higher School of Economics, Nizhny Novgorod, Russia.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network flows: theory, algorithms, and applications. Tech. rep., DTIC Document (1988)
2. Beckmann, M., McGuire, C., Winsten, C.B.: Studies in the economics of transportation. Tech. rep. (1956)
3. Bertsekas, D.P.: Nonlinear programming, 2nd edn. Athena Scientific, Belmont (1999)
4. CPLEX, I.I.: V12. 1: Users manual for CPLEX. Int. Bus. Mach. Corp. **46**(53), 157 (2009)
5. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959)
6. Drud, A.S.: Conopt a large-scale GRG code. ORSA J. Comput. **6**(2), 207–216 (1994)
7. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Nav. Res. Logist. Q. **3**(1–2), 95–110 (1956)
8. Friesz, T.L., Bernstein, D., Smith, T.E., Tobin, R.L., Wie, B.: A variational inequality formulation of the dynamic network user equilibrium problem. Oper. Res. **41**(1), 179–191 (1993)

9. Fukushima, M.: A modified frank-wolfe algorithm for solving the traffic assignment problem. Transp. Res. Part B Methodol. **18**(2), 169–177 (1984)
10. Larsson, T., Patriksson, M.: Side constrained traffic equilibrium models analysis, computation and applications. Transp. Res. Part B Methodol. **33**(4), 233–264 (1999)
11. Rosenthal, R.E.: Gams—a user's guide. GAMS Development Corporation, Washington, DC (2004)
12. Seref, O., Ahuja, R.K., Orlin, J.B.: Incremental network optimization: theory and algorithms. Oper. Res. **57**(3), 586–594 (2009)
13. Sheffi, Y.: Urban transportation networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice-Hall, Inc., Englewood Cliffs (1984)
14. Skiena, S.: Dijkstra's Algorithm. Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica. Addison-Wesley, Reading (1990)
15. Yin, Y., Lawphongpanich, S.: A robust approach to continuous network designs with demand uncertainty. In: Transportation and Traffic Theory 2007. Papers Selected for Presentation at ISTTT17 (2007)
16. Zhang, L., Lawphongpanich, S., Yin, Y.: An active-set algorithm for discrete network design problems. In: Lam, W.H.K., Wong, H., Lo, H.K. (eds.) Transportation and Traffic Theory 2009: Golden Jubilee, pp. 283–300. Springer, Berlin (2009)