Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 4539324, 11 pages
https://doi.org/10.1155/2018/4539324

Research Article

Road Maintenance Optimization Model Based on Dynamic Programming in Urban Traffic Network

Jie Ma ,[1,2] Lin Cheng ,[1] and Dawei Li [1,3,4]

[1] School of Transportation, Southeast University, Nanjing, China
[2] Department of Engineering, National University of Singapore, E1, No. 08-20, 3 Engineering Drive 2, Singapore
[3] Jiangsu Key Laboratory of Urban ITS, Southeast University, Nanjing, China
[4] Collaborative Innovation Center of Modern Urban Traffic, School of Transportation, Southeast University, Nanjing, China

Correspondence should be addressed to Lin Cheng; gist@seu.edu.cn and Dawei Li; lidawei@seu.edu.cn

Received 11 December 2017; Revised 2 May 2018; Accepted 29 May 2018; Published 28 June 2018

Academic Editor: Paola Pellegrini

Urban road maintenance is an important part of urban traffic management. However, in modern cities, road maintenance work needs to occupy some traffic resources; therefore, unreasonable road maintenance schemes often lead traffic networks to unexpected large-scale congestion. In this paper, a dynamic programming model is proposed in order to minimize the delay caused by road maintenance scheme. This model can obtain a globally optimal maintenance scheme which contains the decisions and sequence for every stage of maintenance. Each stage of this model can be boiled down to a discrete network design problem. This model helps make suggestions for the traffic managers with the request of minimizing the delay caused by the maintenance scheme. This paper uses two examples to illustrate this method, one is a small-scale Nguyen-Dupuis network, and the other one is a larger scale Sioux-Falls network.

1. Introduction

After the construction of urban network, the maintenance of urban roads is an important part of urban traffic management. As urban roads are often combined by a large number of travel demands and urban road maintenance takes up a lot of road resources, it is reasonable to develop urban road maintenance schemes.

Many departments of transportation (DOTs) use pavement management systems (PMS) to determine when to maintain the pavements. Using PMS can reduce the total financial costs of the agency; however, it still cannot deal with the delay problems efficiently. In addition, many researchers have begun to focus on other constraints of road maintenance, such as reducing environment pollutes, like greenhouse gas (GHG) [1–3].

The design of road maintenance concerns about multiple objectives. Multicriteria optimization in pavement management systems (PMS) has been used to incorporate heterogeneous objectives simultaneously [4–7]. The studies of road maintenance usually involves many fields, for example, (a) pavement planning [8]; (b) road infrastructure quality [9, 10]; (c) budgets and investments [11, 12]; (d) transportation infrastructure vulnerability [13]; (e) pavement management [14]; (f) link restriction [15]; (g) fuel efficiency [16–19]; (h) GHG emissions [20, 21]; (i) pavement design [22]; (j) separating trucks from passenger vehicles [23–25]; (k) road maintenance in extreme climate [26]; and (l) maintenance equipment [27].

With the development of cities, the urban traffic network becomes more and more complex, and the scale of urban traffic network is also growing. The urban traffic network provides daily services for a large number of travel demand and traffic users. Therefore, an unreasonable road maintenance scheme may result in large-scale traffic congestion and bring huge losses to the traffic network users. In addition, the budget of road maintenance is an important aspect of limiting the progress of road maintenance. Urban traffic network managers often need to develop optimal maintenance schemes according to the limited budget and thus need a more flexible optimization model.

To solve this problem, we propose a dynamic programming model and integrate in the proposed model a subproblem of identifying the best combination of roads to be maintained simultaneously. This subproblem considers the limited budget and the number of links to be maintained simultaneously. This subproblem is indeed a discrete network design problem (DNDP) which has been discussed in many researches. DNDP is usually formulated as a bilevel mixed-integer nonlinear programming (MINLP) which is difficult to solve. The difficulty derives from both the nonlinearity and the nonconvexity which lead it to an NP-hard problem. LeBlanc [28] proposed a branch-and-bound based method to solve DNDP. However, this method can hardly solve large-scale problems. Many heuristic algorithms were proposed to trade off the accuracy and the computational time [29–31]. In order to reduce the computational time, by using a substitution for cost function, Poorzahedy and Turnquist [32] developed a single-level model to approximate the original DNDP. Due to the property of nonlinearity and nonconvexity, the global optimal solution of DNDP is difficult to obtain. Recently, Wang DZW and Lo HK [33] transformed DNDP into a single-level programming and then linearized it into a mixed-integer linear programming (MILP). The global optimum can be obtained and the computational time relies on the linearization scheme. Wang et al. [34] proposed two global optimization solution methods for DNDP; one is SO-relaxation method and the other one is UE-reduction method. Both of them can figure out global optimum and reduced the computational time greatly. However, few published researches on DNDP concerned about the optimal maintenance scheme, although there is strong connection between DNDP and the subproblem of optimal maintenance scheme which will be illustrated in this paper.

Conventionally, DNDP is employed to optimize the topology of a traffic network [28, 34–36]. It can be used to describe the problem whether to add a new link or whether to expand the capacity of existing links in the network. However, in the problem of optimal maintenance scheme, DNDP can be used to temporarily close the best combination of link(s) in every stage of the optimal maintenance scheme. Different from the most general DNDP in literature, the subproblem, which is in the framework of DNDP, does not improve the capacity of the network at all. In fact, it reduces both the capacity of certain link(s) and the accessibility of the network temporarily for the purpose of maintenance. Besides, after each stage of maintenance is completed, the capacity of the maintained link(s) restores as before. In other words, as long as the links are not being maintained, their capacity remains unchanged. The special property differentiates our problem from the general DNDP. What is more, actually, when formulating the proposed model and solving the problem in this paper, we wisely make use of this property. We interpret this property as a guarantee of the non-aftereffect requirement of dynamic programming (DP) so that we can propose a DP for obtaining the optimal maintenance scheme. If this property was ignored, an optimal maintenance scheme could not be obtained like that. DP was first proposed and systematically illustrated by Bellman [37, 38]. Obviously, since DP provides recurrence formulas for obtaining the globally optimal solution, the global optimum of problem can be guaranteed.

This paper provides a reliable model to optimize the maintenance schemes for urban traffic network. This model allows the managers of urban networks to freely set the number of links to be maintained simultaneously according to some limits, for example, the financial budget and the schedule of maintenance. Through this model, we can obtain a road maintenance strategy aiming at minimizing the total generalized costs of the urban traffic network and minimize the influence of the maintenance on road users and provide guidance for urban road network managers. All in all, the main innovations of this paper are as follows:

(1) This paper presents a dynamic programming model to optimize the urban road maintenance scheme, which aims at minimizing the total cost of the network and the degree of influence of road maintenance on users. By doing this, the decisions and sequence of each stage of the maintenance can be obtained.

(2) This paper describes three critical components of the dynamic programming model: optimal substructure, state transition equation, and terminal condition. By utilizing the property of this problem, the paper splits each stage of this model into a subproblem which is a discrete network design problem (DNDP). By doing this, this model allows traffic managers to freely set the number of roads to be simultaneously maintained according to their financial budget, schedule of maintenance, etc.

(3) Two global optimization methods are introduced to solve the subproblem. The algorithm of the subproblem is optimized based on the SO-relaxation method. This algorithm can improve the computational efficiency. With its help, the optimal maintenance scheme can be obtained quickly. Two example networks are tested.

The rest of this article is as follows. Section 2 introduces the principle of user equilibrium (UE), which is widely used in traffic assignment problem (TAP) and reminds us that the system cost at UE state cannot be lower than that at System Optimization (SO) state. Section 3 describes the method of optimizing urban road maintenance schemes and its algorithm. This method is validated by testing two examples in Section 4. We use a small-scale Nguyen-Dupuis network to verify the feasibility of this method. A larger scale Sioux-Falls network shows the efficiency of the algorithm.

2. User Equalization

In this section, we introduce the principle of user equilibrium (UE). Besides, in this paper, for exposure reasons, we also employ UE in the traffic assignment problem (TAP).

All users in the network want to select the shortest path to reach the destination as far as possible, which leads the users with the same origin and destination (OD pair) to choose the same routes and results in link congestion and travel time increasing. Therefore, when the travel time increases to a certain extent, this path will no longer be the shortest path, and the users will turn to other feasible paths. As a result of choice behavior, eventually the entire network achieves a stable state, that all used paths have the same travel costs. No one can reduce their travel costs by unilaterally change his/her choice of path. Such state is called the user equilibrium state. Obviously, according to the definition of user equilibrium, the user equilibrium state has the property of spontaneity.

Besides, we should note that, when the maintenance happens, the users will respond to this change of the network within a certain time. The time depends on the level of information and perception. In modern traffic networks, we can use many existing techniques, such as information center, navigators, Connected Vehicles (CV), and Intelligent Transportation System (ITS), to obtain the real-time information about the road circumstance of a network. If the level of information and perception is not high enough, only a small part of users will change their route choice. Others may change after a certain time afterwards. But at last, maybe after a long time, all the users will reach the UE state again. Otherwise, if the level of information and perception is high enough, most users can receive the information within a very short time so that they can respond within a short time. In this case, the users can reach the UE state again quickly and probably before the maintenance is finished. Because of the increasing level of information and perception in modern cities, the UE state will be reached more and more quickly and UE models will also work better and better.

For a general urban network, because of its large scale, it is generally a complex network with lots of users. In general, the UE state keeps the users in an efficient state. However, due to the existence of the Braess paradox in networks, a poor traffic state may come out. It makes the total travel time longer than that of an SO state and all users need to spend equal but longer travel time.

3. Methodology

In this section, we propose a model to solve the problem of optimal maintenance scheme.

Before proposing the model, we first introduce some notations that may be involved in the model proposed in this paper.

The following notations are used in the formulation:

A: the set of all link in the traffic network

c_p^w: the travel time on path p between OD pair $w \in W$

$\mathbf{f}$: a vector defined as $\mathbf{f} = [f_p^w], \forall p \in R_w, w \in W$

f_p^w: the flow on path p between OD pair $w \in W$

N: the number of links in the traffic network

q^w: the traffic demand between OD pair $w \in W$, which is given as a constant

R_w: the set of all paths between OD pair $w \in W$

t_a: the travel time on link $a \in A$

$\mathbf{u}$: a vector defined as the network design decision, $\mathbf{u} = [u_a], \forall a \in A$

u_a: a binary variable which indicates whether a link is a potential link which needs to be maintained: $u_a = 1$ if link a needs to be maintained; otherwise $u_a = 0$

V: the number of links that manager wants to maintain simultaneously in the network

W: the set of all OD pairs in the traffic network

$\mathbf{x}$: a vector defined as $\mathbf{x} = [x_a], \forall a \in A$

x_a: the traffic flow of link $a \in A$

$\Omega_\mathbf{f}$: the set of all feasible path flows $\mathbf{f}$, $\Omega_\mathbf{f} = \{\mathbf{f} \mid \sum_{p\in R_w} f_p^w = q^w,\ f_p^w \geq 0,\ \forall w \in W,\ p \in R_w\}$

$\Omega_\mathbf{u}$: the set of all feasible network design decisions $\mathbf{u}$, $\Omega_\mathbf{u} = \{\mathbf{u} \mid u_a = \{0, 1\},\ \forall a \in A\}$

$\Omega_\mathbf{x}$: the set of all feasible link flows $\mathbf{x}$, $\Omega_\mathbf{x} = \{\mathbf{x} \mid x_a = \sum_{w\in W}\sum_{p\in R_w} \delta_{a,p}^w f_p^w,\ \forall a \in A,\ \mathbf{f} \in \Omega_\mathbf{f}\}$

$\boldsymbol{\delta}$: the link-path incidence matrix, $\boldsymbol{\delta} = [\delta_{a,p}^w], \forall w \in W$, $a \in A$, $p \in R_w$, where $\delta_{a,p}^w = 1$ if link a belongs to path p between OD pair w, and $\delta_{a,p}^w = 0$ otherwise

σ_p^w: a binary variable which indicates whether a path is used: path p is used if $\sigma_p^w = 0$ and is not used if $\sigma_p^w = 1$

The purpose of our maintenance schemes is to minimize the delay caused by the maintenance work. Thus, we first propose a subproblem which is an optimization problem as follows. This optimization problem is to find out the best combination of link(s) to be maintained simultaneously. The objective function of the optimization is

$$\min Z = \sum_{a\in A} x_a t_a - x_a^* t_a^* \tag{1}$$

where x_a^* represents the flow on link a before being maintained and t_a^* represents its travel time.

One should note that, in (1), the second term $x_a^* t_a^*$ is a constant. Thus, (1) can be transformed into

$$\min Z = \sum_{a\in A} x_a t_a \tag{2}$$

When making maintenance schemes, due to the maintenance costs and duration, etc., it is often needed to limit the number of the links simultaneously maintained. In addition, too many links maintained at the same time may cause a sharp increase in travel time. Therefore, it is sometimes important to appropriately select the number of links to be maintained at the same time. These factors are taken into account in the design of our model so that the traffic managers can set the number of links to be maintained at the same time according to their own needs. By doing this, it can also help managers make decisions by comparing the results of setting different numbers of links simultaneously maintained.

$$\begin{gathered}\sum_{a\in A} u_a = N - V \\ u_a \in \{0, 1\}, \quad a \in A\end{gathered} \tag{3}$$

Constraint (3) allows the traffic manager to freely set the number of links to be maintained simultaneously. V represents the number of simultaneously maintained links, and this value can be freely set by the manager. For example, if the number of links that manager wants to maintain simultaneously is 2, we set $V = 2$. u_a is a binary variable, which represents whether link a needs to be maintained (e.g., if link a needs to be maintained, $u_a = 1$; otherwise, $u_a = 0$).

Note: (3) actually indicates that all links in the network are needed to be maintained. However, the proposed model is also available for the case where a scope of links is given.

In most cases, the links needed to be maintained are only a small part of the whole network. Thus, we do not need to consider all the links. Accordingly, we modify (3) into (3a):

$$\begin{gathered}\sum_{a\in A'} u_a = N' - V \\ u_a = 0, \quad a \in A \setminus A' \\ u_a \in \{0,1\}, \quad a \in A'\end{gathered} \tag{3a}$$

where A' denotes the set of links needed to be maintained; N' denotes the number of links needed to be maintained.

To illustrate (3a) more clearly, here, we make use of the example of Nguyen-Dupuis (ND) network shown in later section. For example, there are 5 links needed to be maintained (e.g., in the ND network, 5-9, 7-8, 9-10, 10-11, and 12-8), and the manager wants 2 links to be maintained simultaneously due to the funds and schedule limit. Then, we set N'=5, A' = {5-9, 7-8, 9-10, 10-11 and 12-8} and V=2. The other constraints and objective function keep unchanged.

(3a) is an adaptation for the case where a scope of links to be maintained is given. Details can be seen in later section.

$$t_a = t_{a,0}\left[1 + b\left(\frac{x_a}{y_a}\right)^m\right] + u_a E, \quad a \in A \tag{4}$$

where y_a is the capacity of link a; m is a parameter which can be set according to actual cases and often be set as 4 in general; E is the parameter which represents the performance change due to the maintenance work; the other parameters are the same as above.

Constraint (4) indicates that when link a is maintained, its cost is increased due to the change of its capacity. For example, when link a needs to be fully enclosed due to the maintenance scheme so that no vehicles can pass through, in this case, E is an infinite large number. When only parts of link a need to be closed, although it can still meet some parts of the traffic demand, but its capacity is limited, which results in the increased travel time. In this case, E can be set as a corresponding positive number in practice.

$$\begin{gathered}L \cdot \sigma_p^w + \varepsilon \le f_p^w \le M \cdot \left(1 - \sigma_p^w\right) \\ L \cdot \sigma_p^w \le c_p^w - \pi^w \le M \cdot \sigma_p^w \\ c_p^w - \pi^w \ge 0 \\ \sigma_p^w \in \{0,1\} \\ \forall p \in R_w, \quad w \in W\end{gathered} \tag{5}$$

Constraint (5) is a classical representation of Wardrop's principle which is also known as the UE principle in traffic assignment problems (TAP) [33].

$$\begin{gathered}d^w = \sum_{p\in R_w} f_p^w, \quad w \in W \\ x_a = \sum_{w\in W}\sum_{p\in R_w} \delta_{a,p}^w f_p^w, \quad a \in A \\ c_p^w = \sum_{a\in A} \delta_{a,p}^w t_a, \quad p \in R_w,\ w \in W \\ x_a \ge 0, \quad a \in A \\ f_p^w \ge 0, \quad \forall p \in R_w,\ w \in W\end{gathered} \tag{6}$$

Constraint (6) is the nonnegative constraint and the flow conservation constraint.

We rewrite the above mixed-integer nonlinear programming (MINLP) as follows:

[MINLP]

$$\min Z = \sum_{a\in A} x_a t_a \tag{7}$$

$$\begin{gathered}\sum_{a\in A} u_a = N - V \\ u_a \in \{0,1\}, \quad a \in A\end{gathered} \tag{8}$$

$$t_a = t_{a,0}\left[1 + b\left(\frac{x_a}{y_a}\right)^m\right] + u_a E, \quad a \in A \tag{9}$$

$$\begin{gathered}L \cdot \sigma_p^w + \varepsilon \le f_p^w \le M \cdot \left(1 - \sigma_p^w\right) \\ L \cdot \sigma_p^w \le c_p^w - \pi^w \le M \cdot \sigma_p^w \\ c_p^w - \pi^w \ge 0 \\ \sigma_p^w \in \{0,1\} \\ \forall p \in R_w, \quad w \in W\end{gathered} \tag{10}$$

$$\begin{gathered}d^w = \sum_{p\in R_w} f_p^w, \quad w \in W \\ x_a = \sum_{w\in W}\sum_{p\in R_w} \delta_{a,p}^w f_p^w, \quad a \in A \\ c_p^w = \sum_{a\in A} \delta_{a,p}^w t_a, \quad p \in R_w,\ w \in W \\ x_a \ge 0, \quad a \in A \\ f_p^w \ge 0, \quad \forall p \in R_w,\ w \in W\end{gathered} \tag{11}$$

Note that the only thing which makes [MINLP] a nonlinear model is (9). Once the nonlinearity of (9) is tackled, both of the constraints and the objective function of [MINLP] are linearized. Then, [MINLP] becomes a mixed-integer nonlinear programming (MILP) which is known as a global optimization model. Here, we employ a linearization method which is proposed by Wang et al. [33]. For each link a, we choose a series of $K_{a,n}$ to partition the feasible domains of x_a into many small regions. In each region $[n, n+1]$, $K_{a,n} \le x_a - K_{a,n} < K_{a,n+1}$. Then, we linearize the first term on the right of (9) (the second term is already a linear one) by the following linear constraints:

$$\begin{gathered} L \cdot \zeta_{a,n} \le x_a - K_{a,n} \le U \cdot (1 - \zeta_{a,n}) - \varepsilon \\ \zeta_{a,n} = \zeta_{a,n+1} - \zeta_{a,n} \\ L \cdot (1 - \zeta_{a,n}) \le t_a - a_{a,n} x_a \le U \cdot (1 - \zeta_{a,n}) \\ \zeta_{a,n} \in \{0, 1\} \end{gathered} \tag{9a}$$

where $a_{a,n} = (dt_a/dx_a)|(K_{a,n})$ which is a known coefficient, since $K_{a,n}$ is prespecified.

By doing this, [MINLP] becomes an MILP which has globally optimal solution. However, the computing efficiency of this model strongly relies on the linearization scheme. The details and the proofs can be seen in literature [33, 39]. If the high resolution of the linearization scheme is required, the computing time will become very long. This is very uneconomic especially when there are only a few links needed to be maintained. Thus, we introduce another efficient method which is also a global optimization.

We note that some of the constraints of [MINLP] are for solving TAP. Therefore, taking the advantage of them, we transform [MINLP] into a bilevel model for the road maintenance problems.

The upper-level optimization indicates that traffic managers aim to minimize the total travel time in maintenance problems by setting appropriate binary decision variables $\mathbf{u}$.

[OP: Original Problem]

$$\min_{\mathbf{u},\mathbf{x}} \sum_{a \in A} x_a t_a \tag{12}$$

Subject to

$$\sum_a u_a = N - V, \quad \forall a \in A \tag{13}$$

$$u_a \in \{0, 1\}, \quad \forall a \in A \tag{14}$$

Constraints (13)-(14) are equal to constraint (8).

The lower-level optimization is user equilibrium traffic assignment problem. When the network design decision $\mathbf{u} \in \Omega_{\mathbf{u}}$ changes, the traffic flow $\mathbf{x}$ over the network may change. We assume that the demand for travel is given and fixed, and users follow the user equilibrium principle:

[UE]

$$\min \sum_{a \in A} \int_0^{x_a} t_a(x)\,dx \tag{15}$$

Subject to

$$\sum_{p \in R_w} f_p^w = q_w, \quad \forall w \in W \tag{16}$$

$$x_a = \sum_{w \in W} \sum_{p \in R_w} \delta_{a,p}^w f_p^w, \quad \forall a \in A \tag{17}$$

$$f_p^w \ge 0, \quad \forall p \in R_w, \ \forall w \in W \tag{18}$$

Formulas (15)– (18) are equal to constraint (10).

One can note that both the constraints and the objective function of [UE] are convex, which means [UE] always has a unique solution. After obtaining the solution of [UE] for each decision $\mathbf{u}$, we take it into [OP] and compare the value of objective function. After enumerating all feasible decisions, we can figure out the best solution of $\mathbf{u}$ that makes the travel time of the network minimum. Obviously, this is an enumeration method. Thus, the global optimum can be inherently guaranteed (although the solution may be not unique).

Since this is an enumeration method, it is efficient enough only in the cases where a small number of links need to be maintained. However, going through this enumeration process to calculate all the combination of more links may be absurd and occupy large amount of calculating resource. So, we present an approach to avoid the fussy calculation.

Since the number of iterations of [UE] depends on [OP] and the number of combinations of links is exponential to the number of links in [OP], [OP] is the key factor that leads to the huge computational complexity. Considering that [OP] is an MINLP problem, the calculation of [OP] itself is also a complex problem. Therefore, we try to modify the algorithm through optimizing [OP].

We use SO-relaxation method to optimize [OP]. The SO-relaxation method was first mentioned by Wang et al. [34]. This method works because of the Braess paradox which makes the total cost of a network at UE not lower than that at SO. A research of using this method to analyze Braess paradox can be seen in Ma [15].

We propose the following relaxed problem (RP) to relax the range of the variable $\mathbf{x}$ of [OP]:

[RP]

$$\min_{\mathbf{u} \in \Omega_{\mathbf{u}}, \mathbf{x} \in \Omega_{\mathbf{x}}} \sum_{a \in A} x_a t_a(x_a) \tag{19}$$

where $\mathbf{x} \in \Omega_{\mathbf{x}}$ and other constraints of [RP] are the same as [OP].

However, in [OP]:

$$\mathbf{x} \in \Omega_{\mathbf{x}}^{UE}(\mathbf{u}) \tag{20}$$

where

$$\Omega_{\mathbf{x}}^{UE}(\mathbf{u}) = \arg \min_{\tilde{\mathbf{x}} \in \Omega_{\mathbf{x}}} \sum_{a \in A} \int_0^{\tilde{x}_a} t_a(x)\,dx \tag{21}$$

Obviously, $\Omega_{\mathbf{x}}^{UE}(\mathbf{u}) \subset \Omega_{\mathbf{x}}$; that is the reason why we call [RP] a relaxed problem of [OP].

One can note that, since [RP] is a relaxed problem of [OP], which means the range of variables in [RP] is larger than that in [OP]. Therefore, the optimal solution of [OP] should be the optimal solution of [RP] first. That is to say, the objective function value (total cost) of [RP] of the maintenance decision provides a lower bound for that of [OP] under this decision. The objective function value of [OP] of any maintenance decision cannot be lower than that of [RP] of the same decision. Therefore, we can first minimize the objective function of [RP] to obtain the best solution $\mathbf{u}*$ for [RP] and calculate the objective function value of [OP] of $\mathbf{u}*$. Then we compare it with the objective function value of other solutions for [RP] (i.e., other maintenance decisions). If the objective function value of [RP] of other decisions is larger than that of [OP] of $\mathbf{u}*$, then $\mathbf{u}*$ is the optimal solution, and [OP] of other decisions does not need to be solved. What is more, we note that [RP] is actually a single-level model. That means we can obtain its optimal solution easily. Thus, we can sort its solutions for different decisions orderly from minimum to maximum by kicking out former solutions from $\Omega_{\mathbf{u}}$(with the following algorithm). By doing this, once the objective function value of a specific solution for [RP] is larger that of $\mathbf{u}*$, all later solutions also need not to be tested, orderly from minimum to maximum

By the above method, we convert a large number of calculations of [OP] into one calculation of [RP]. The [RP] problem can be transformed into a mixed-integer linear problem by multicommodity flow (MCF) [34], which is easy to calculate by personal computer.

Based on above analysis, we propose an algorithm to solve this bilevel problem as follows:

Step 0: define a set $\overline{\Omega}_{\mathbf{u}} = \varnothing$ which will include all the generated solutions for generating the cuts through (21). Define the upper bound: $UB = +\infty$, incumbent optimal solution $\mathbf{u}^{opt}$.

Step 1: solve [RP] based on the following constraint.

$$\sum_{a \in A} \left[\left(1 - u_a\right) \overline{u}_a + \left(1 - \overline{u}_a\right) u_a \right] \geq 2, \quad \forall \overline{\mathbf{u}} \in \overline{\Omega}_{\mathbf{u}} \tag{22}$$

Constraint (22) excludes all the solutions in $\overline{\Omega}_{\mathbf{u}}$, so that the feasible region becomes $\Omega_{\mathbf{u}} \setminus \overline{\Omega}_{\mathbf{u}}$. Constraint (22) ensures the calculating solution $\mathbf{u}$ is not a calculated one $\overline{\mathbf{u}}$. Case 1.1: if this problem has no solutions; that is to say we have listed all feasible solutions and hence $\mathbf{u}^{\text{opt}}$ is the optimal solution and stop. Case 1.2: otherwise, obtain the provisional optimal solution denoted by $\mathbf{u}^*$, and the provisional optimal value denoted by Obj_{RP} under constraint (22). Case 1.2.1: if $Obj_{RP} \geq UB$, $\mathbf{u}^{\text{opt}}$ is the optimal solution and stop. Case 1.2.2: otherwise (if $Obj_{RP} < UB$), go to Step 2.

Step 2: fix the value of $\mathbf{u}$ at $\mathbf{u}^*$ and solve [UE] and obtain the UE link flow solution denoted by $\mathbf{x}^*$. Case 2.1: if

$$UB \leq \sum_{a \in A} x_a^* t_a \left(x_a^*\right) \tag{23}$$

then $\mathbf{u}^*$is not a better solution than $\mathbf{u}^{\text{opt}}$. Then we set $\overline{\Omega}_{\mathbf{u}} = \overline{\Omega}_{\mathbf{u}} \cup \{\mathbf{u}^*\}$, and go to Step 1. (Case 2.2: otherwise ($\mathbf{u}^*$is a better solution than $\mathbf{u}^{\text{opt}}$), set

$$UB = \sum_{a \in A} x_a^* t_a \left(x_a^*\right) \tag{24}$$

$$\mathbf{u}^{\text{opt}} = \mathbf{u}^* \tag{25}$$

(Case 2.2.1: if $Obj_{RP} \geq UB$, $\mathbf{u}^{\text{opt}}$ is the optimal solution and stop. Case 2.2.2: otherwise ($Obj_{RP} < UB$), set $\overline{\Omega}_{\mathbf{u}} = \overline{\Omega}_{\mathbf{u}} \cup \{\mathbf{u}^*\}$ and go to step 1.

By employing [RP], the computational efficiency has been greatly improved. Lots of repeated calculations of UE are replaced by one SO process which is actually a single-level programming whose running time is negligible compared with that of [OP].

So far, we have introduced two methods whose solutions are globally optimal. The former one is efficient when links in the network are not too many. This is because more links lead to more binary variables for the linearization scheme. And the latter one is extremely suitable for the cases where there are only a few links to be maintained in a large network.

One should note that both of the above methods only have dealt with one problem: given the number of links that would be maintained at the same time, properly selecting links to make best choice according to the generalized costs (monetary costs and traffic congestion). However, our purpose is to obtain the optimal maintenance scheme (i.e., given N' links to be maintained, find the optimal *order* of maintains). The above two methods cannot achieve this goal directly but play an important role in achieving it.

We find that, given N' links to be maintained, the optimal maintenance scheme establishes on both the optimal maintenance scheme when given $N' - V$ links to be maintained and the best combination of V links. Take the advantage of this rule, we propose a dynamic programming (DP) to describe our problem and then achieve our goal.

Before we give out the [DP], we first explain our problem with a diagram. Suppose we have N'=6 links to maintain, and we can maintain V=2 links at a stage. We brief the links as $\{1, 2, 3, 4, 5, 6\}$. A three-stage maintenance scheme can be simplified as a directed acyclic graph as shown in Figure 1.

In Figure 1, there are three stages each of which represents one stage of the maintenance scheme. For example, Stage 1 shows there are 4 links which remain to be maintained, which is also briefed as $F(2)$ in [DP]. Correspondingly, Stages 2 and 3 are briefed as $F(4)$ and $F(6)$ in [DP], respectively. In Stages 1 and 2, there are 15 nodes for each. The number in each node denotes the links remained to be maintained. The number on each link denotes the cost on this link which can be determined by [OP]. For example, the number *56* on the up-left link denotes that the cost on this link equals the objective function value of [OP] when the maintained links are *5* and *6*. For the space limitations, we omit the numbers on the links between Stage 1 and Stage 2. Note, in the cases where N' cannot be divided by V or V equals other values, the directed acyclic graph is similar to Figure 1. We omit the diagrams of these cases due to space limitations; however, these cases will be carefully considered in the proposed [DP].

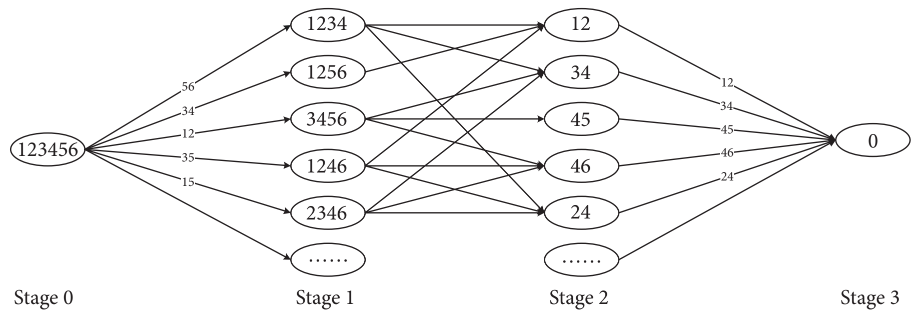

FIGURE 1: A directed acyclic graph of the three-stage maintenance scheme.

From the above explanation, we transform the problem into a classic shortest path problem which can be easily solved by many existing algorithms [40, 41] whose running time is negligible compared with [UE]. By these algorithms, actually, we can find the shortest path between arbitrary two nodes. Obviously, the optimal maintenance scheme in Figure 1 can be obtained by finding the shortest path from Stage 0 to Stage 3. Thus, in [DP], the optimal substructure $F(N'-1), F(N'-2), \ldots, F(N'-V)$ is actually solved by the famous Dijkstra algorithm for the shortest path problem. And for exposure reasons, in [DP], we expand it explicitly.

Now we have done all the preliminary work for proposing a dynamic programming for our problem. Since V is a freely set number, the cases where N' cannot be divided by V must be very common. However, this problem will be solved in [DP].

[DP]. To describe this problem, we give the following definitions:

$F(N') \triangleq$ given N' link(s) to be maintained, the generalized cost of the optimal maintenance scheme

$G_V(N') \triangleq$ given $F(N')$ and N' link(s) have been maintained, the generalized cost of the best combination of V link(s)

N' denotes the number of links to be maintained; V denotes the number of links to be maintained simultaneously. Obviously, if $N' \le 0$, $F(N') = 0$; if $V \le 0$, $G_V(N') = 0$.

The state transition equation is shown as follows:

$$\begin{aligned} F\left(N'\right) = \min\{&F\left(N'-1\right) \\ &+ G_1\left(N'-1\right), F\left(N'-2\right) \\ &+ G_2\left(N'-2\right), \ldots, F\left(N'-V\right) + G_V\left(N'-V\right)\} \end{aligned} \tag{26}$$

The optimal substructure of [DP] is $F(N'-1), F(N'-2), \ldots, F(N'-V)$. And the terminal condition of [DP] is $F(1)$.

Note, due to the definition of $F(N')$ and $G_V(N')$, when $N' = 1$

$$F(1) = G_1(0) \tag{27}$$

(26) and (27) show that not only the state transition equation is relative with $G_V(N')$, even the terminal condition can be determined only by $G_V(N')$. That is to say, once we get $G_V(N')$, the optimal maintenance scheme $F(N')$ should be obtained. What is more, one should note that $G_V(N')$ is exactly the results obtained by [MINLP] and [OP]. That is why we introduce these two methods in former sections.

The calculation of $G_V(N')$ involves the flow pattern of the network which, in this paper, is determined by link performance function and UE. Note that, whether before or after the links are maintained (as long as they are not being maintained currently), the link performance function does not vary, so the flow pattern of the network remains unchanged. This ensures the non-aftereffect property required by dynamic programming.

To illustrate [DP] more clearly, we set $N' = 5$, $V = 2$. The up-bottom process of [DP] is

$$F(5) = \min\{F(4) + G_1(4), F(3) + G_2(3)\}; \tag{28}$$

$$F(4) = \min\{F(3) + G_1(3), F(2) + G_2(2)\}; \tag{29}$$

$$F(3) = \min\{F(2) + G_1(2), F(1) + G_2(1)\}; \tag{30}$$

$$F(2) = \min\{F(1) + G_1(1), G_2(0)\}; \tag{31}$$

$$F(1) = G_1(0). \tag{32}$$

In this example, there are 5 links to be maintained. We can obtain the optimal maintenance scheme with a bottom-up process (from (32) up to (28)). First of all, we calculate $G_1(0)$ which means to find the best combination of 1 link when no links have already been maintained. The solution is naturally $F(1)$. Then we calculate $G_1(1)$ and $G_2(0)$, respectively. $G_1(1)$ denotes finding the best link when 1 link (i.e., the link of $F(1)$) has already been maintained. And $G_2(0)$ denotes finding the best combination of 2 links when no links have already been maintained. After comparing $F(1) + G_1(1)$ and $G_2(0)$, the scheme with lower cost is recorded as $F(2)$. The rest process (28)-(30) has similar interpretation. During the whole process, once the current link has already been maintained, it will never be maintained again and will be kicked out of the links to be maintained. That is to say, the calculation time of $G_1(0)$ is always longer than that of $G_1(1)$.

One should also note that $G_V(N')$ is nonadditive (i.e., $G_2(0) \ne G_1(0) + G_1(1)$). That is why we need to compare the two terms in the above process and why the above process

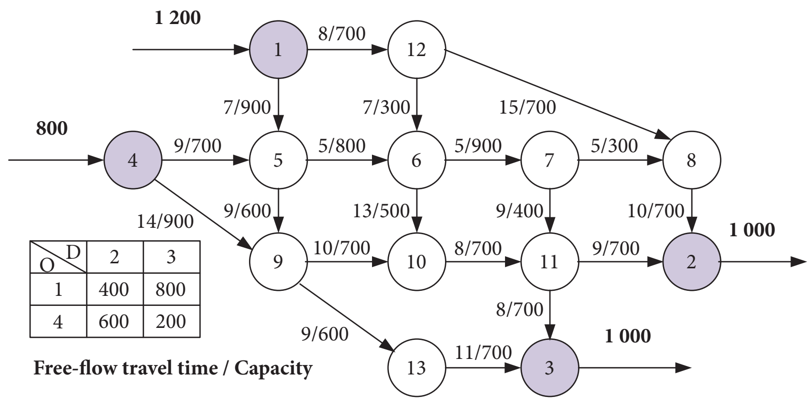

Figure 2: The Nguyen-Dupuis network with its OD pairs and capacities.

Table 1: Free-flow travel time and capacities of the links in Nguyen-Dupuis network.

Link	Link serial	c_a veh/h	$t_{a,0}$	Link	Link serial	c_a veh/h	$t_{a,0}$
1-5	1	900	7	8-2	11	700	10
1-12	2	700	8	9-10	12	700	10
4-5	3	700	9	9-13	13	600	9
4-9	4	900	14	10-11	14	700	8
5-6	5	800	5	11-2	15	700	9
5-9	6	600	9	11-3	16	700	8
6-7	7	900	5	12-6	17	300	7
6-10	8	500	13	12-8	18	700	15
7-8	9	300	5	13-3	19	700	11
7-11	10	400	9				

cannot be broken up any further. According to this bottom-up process, it is easily to see that, for each stage of [DP], only the two terms of $G_1(N'-1)$ and $G_1(N'-2)$ need to be calculated. So the time complexity of [DP] is $o(N' \times V)$, and the spatial complexity is $o(V)$. Both of them are efficient enough. Details of the illustration of [DP] will be present in the next section.

In the next section, we test the proposed algorithm through two numerical examples; one is for the Nguyen-Dupuis network and the other is for a larger scale network, the well-known Sioux-Falls network.

4. Numerical Examples

In order to clearly illustrate the proposed method, we use two different networks as the examples.

4.1. Nguyen-Dupuis Network. We use a personal computer with an Intel(R) Core(TM) i5 4210M @ 2.60 GHz CPU, an 8GB RAM and Windows 8.1 Enterprise operating system (64-bit) to do the numerical test. The model was coded with MATLAB and the lower-level model [UE] is dealt with Convex Combinations Method.

We testify the proposed method in Nguyen-Dupuis network. It has 13 nodes, 19 links, and 4 OD pairs. The diagram of it can be seen in Figure 2.

In this example, the link performance function of each link is the BPR function, and the parameters α, β are 0.15 and 4. The links' free-flow travel times $t_{a,0}$ and their capacities are listed in Table 1. We suppose there are 5 links {5-9, 7-8, 9-10, 10-11, and 12-8} to be maintained and expect to obtain a maintenance scheme in which $V=2$.

We apply the proposed method to the Nguyen-Dupuis network. The test results can be seen in Table 2.

For exposure reasons, we show details of the first stage of our maintenance. However, since both of [MINLP] and [RP] are models which have already been optimized, they can only give out the best selection of V links according to the generalized costs (i.e., $G_V(N')$). Here we use [OP] to list all feasible solutions of the first stage in Table 2.

According to the first column $G_1(0)$ in Table 2, we know that, when V=1 and no links have been maintained before, the best selection of link is (9-10). That is to say, $F(1) = G_1(0)$ =9.6664E+03, and the term of $G_1(1)$ is determined as $G_1((9-10))$. Take advantage of the non-aftereffect of this [DP], the total costs of the selections in $G_1((9-10))$ can be copied from those in $G_1(0)$. Thus, $G_1(1) = G_1((9-10))$ =1.0548E+04. The column of $G_2(0)$ shows that the best combination of links is (7-8, 12-8), when V=2 and no links have been maintained before. Compare the terms of $G_2(0)$ =1.0110E+04 and $G_1(0) + G_1(1)$ =2.0214E+04; $F(2) = F((7-8, 12-8))$ =1.0110E+04 can be determined according to (31). Later stages are similar to the first stage; their optimal solutions can be determined in terms of (28)-(30). After obtaining $F(5)$, here we give out the solution of each stage

Table 2: Feasible solutions of the first stage of the optimal maintenance scheme.

	G_1 (0)			G_1 ((9 − 10))	
Order	**Link**	**Cost**	**Order**	**Link**	**Cost**
1	(9-10)	9.6664E+03	1	(10-11)	1.0548E+04
2	(10-11)	1.0548E+04	2	(12-8)	1.7192E+04
3	(12-8)	1.7192E+04	3	(5-9)	2.4606E+04
4	(5-9)	2.4606E+04	4	(7-8)	2.8514E+04
5	(7-8)	2.8514E+04			
			G_2 (0)		
Order	**Link**	**Cost**	**Order**	**Link**	**Cost**
1	(7-8,12-8)	1.0110E+04	6	(5-9, 12-8)	1.7224E+04
2	(9-10, 10-11)	1.0548E+04	7	(9-10, 12-8)	2.1207E+04
3	(5-9, 9-10)	1.2044E+04	8	(7-8, 9-10)	2.5950E+04
4	(12-8, 10-11)	1.6357E+04	9	(5-9, 7-8)	3.7239E+04
5	(5-9, 10-11)	1.6828E+04	10	(7-8, 10-11)	4.7624E+04

Table 3: Optimal maintenance scheme of the Nguyen-Dupuis network.

Stage	Link combination	Cost of the stage	Cost in total
1	(7-8, 12-8)	1.0110E+04	1.0110E+04
2	(5-9, 9-10)	1.2044E+04	2.2154E+04
3	(10-11)	1.0548E+04	3.2702E+04

directly in Table 3. The optimal maintenance scheme of this example consists of three stages in total.

From the test results, we can see that the optimal maintenance scheme is (7-8, 12-8), (5-9, 9-10), and (10-11) in order. One may note that the cost of the 3rd stage is lower than that of the 2nd stage. Why not maintain the link (10-11) at the 2nd stage then the links (5-9, 9-10)? Actually, the order of maintenance does not affect the total costs. If you want, you can even put (10-11) to the 1st stage. The critical factors are the combinations rather than the orders. Reasons have been explained in former sections. It is because the cost of maintaining the links does not vary, no matter whether the other links have been maintained before or not.

4.2. Sioux-Falls Network. We test the computational efficiency of the proposed method with Sioux-Falls network, which has 24 nodes, 76 links, and 528 OD pairs as Figure 3.

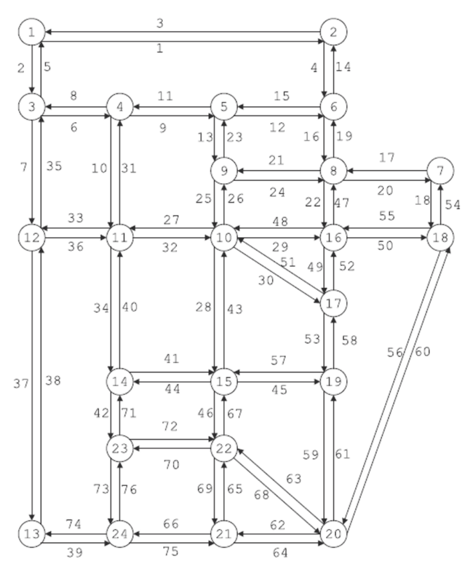

Figure 3: Sioux-Falls network.

As analyzed in the former section, the time complexity of [DP] is $o(N' \times V)$, and all the critical calculation process of [DP] can be boiled down to the calculation of $G_V(N')$. Therefore, it is sufficient enough to consider only the calculation process of $G_V(N')$. As mentioned above, $G_V(N')$ can be determined by both [MINLP] and [RP]. The computational efficiency of [MINLP] mainly relies on the linearization scheme employed. The value of V affects little about the computational efficiency. In [OP], the value of V matters the computational efficiency very much. Actually, the running time of [OP] is exponential to V. However, with the help of [RP], the running time is now greatly reduced. To show this, we reveal the computational efficiency of [RP] when generating $G_V(N')$.

The computational cost of $G_V(N')$ depends on the number of N' and V. A large N' or V may make this problem hard to calculate. To test it in a severe circumstance, we assume a large N' and a large V, respectively.

For a large N', obviously, when N' is given, the times of generating $G_V(N')$ are $G_V(0) > G_V(1) > \ldots > G_V(N')$. What is more, we expand the scope of maintenance to the overall network (i.e., $N' = N$) and test $G_V(0)$ in various cases where V varies from 1 to 3.

In this example, the link travel time functions are BPR functions; the parameters α, β are 0.15 and 4, respectively. The free-flow travel time of each link is the same as the input data proposed by Leblanc [28]. The links' capacities and traffic

Table 4: Test results of the calculation of $G_V(0)$ in Sioux-Falls network.

V	Number of feasible solutions	Best combination	Running time (s)
1	76	(51)	152
2	2850	(30, 51)	166
3	70300	(30, 31, 51)	502

demands follow those proposed by Wang et al. [33]. Test results are shown in Table 4.

Along with V increases, the number of feasible solutions of $G_V(0)$ increases sharply. However, due to [RP], the running time remains affordable. Note that, in this example, we have considered an extremely severe circumstance that $N' = N$. This means the overall network needs to be maintained. Normally, the number of links to be maintained does not exceed 10. Therefore, we test with a normal N' and a large V subsequently.

In the case where 10 links are needed to be maintained, even if V=5 (the number of feasible solutions is $N'!/V!(N' - V)!$. When V>5, the number of feasible solutions decreases, so does the computational cost); it only takes around 160 s to obtain the optimal solution.

5. Conclusions and Research Extensions

In this paper, we propose a dynamic programming model [DP] to optimize the urban road maintenance scheme. This model allows maintenance managers to freely set the number of links to be simultaneously maintained in each stage of the maintenance scheme. Through this model, the decisions and sequence of each stage of the maintenance can be obtained.

We describe three critical components of the dynamic programming model: optimal substructure, state transition equation, and terminal condition. By utilizing the property of this problem, the paper splits each stage of this model into a subproblem which is a discrete network design problem (DNDP).

The subproblem can be solved by two global optimization methods [MINLP] and [OP] whose computational efficiency matters the running time of [DP]. The computational efficiency of [MINLP] relies on the linearization scheme employed. And [OP] can be optimized by SO-relaxation method. With its help, the running time of the proposed model is reduced sharply and the optimal maintenance scheme can be obtained quickly.

We test this model in two networks, one is a small-scale Nguyen-Dupuis network, and the other one is a larger scale Sioux-Falls network. Through these two examples, both the validity and the efficiency of this method are verified. This method can be applied in large-scale networks.

In this paper, the maintenance time is described as stages. In future research, we will consider unsynchronized maintenance time which will definitely make the problem more complex but more practical. However, since maintenance is a simple work in our modern cities. In general, only a few days are needed. And the times consumed for maintaining different links usually do not differ too much. Thus, the proposed model is also practical to some degree. It can help the urban traffic managers to figure out optimal road maintenance schemes effectively and efficiently.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (nos. 51608115, 51578150, and 51378119), Projects of International Cooperation and Exchange of the National Natural Science Foundation of China (no. 51561135003), the Natural Science Foundation of Jiangsu Province (no. BK20150613), and the Scientific Research Foundation of the Graduate School of Southeast University (no. YBJJ1840).

References

[1] N. J. Santero, J. Harvey, and A. Horvath, "Environmental policy for long-life pavements," *Transportation Research Part D: Transport and Environment*, vol. 16, no. 2, pp. 129–136, 2011.

[2] N. J. Santero, E. Masanet, and A. Horvath, "Life-cycle assessment of pavements. Part I: Critical review," *Resources, Conservation & Recycling*, vol. 55, no. 9-10, pp. 801–809, 2011.

[3] N. J. Santero, E. Masanet, and A. Horvath, "Life-cycle assessment of pavements Part II: Filling the research gaps," *Resources, Conservation & Recycling*, vol. 55, no. 9-10, pp. 810–818, 2011.

[4] T. F. Fwa, W. T. Chan, and K. Z. Hoque, "Multiobjective optimization for pavement maintenance programming," *Journal of Transportation Engineering*, vol. 126, no. 5, pp. 367–374, 2000.

[5] J. R. Mbwana, "A framework for developing stochastic multi-objective pavement management systems," in *Proceedings of the First Road Transportation Technology Transfer Conference in Africa 2001*, 2001.

[6] Z. Li and K. C. Sinha, "Methodology for multicriteria decision making in highway asset management," *Transportation Research Record*, no. 1885, pp. 79–87, 2004.

[7] Q. Bai, A. Ahmed, Z. Li, and S. Labi, "A hybrid pareto frontier generation method for trade-off analysis in transportation asset management," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 3, pp. 163–180, 2015.

[8] X. Espinet, A. Schweikert, N. van den Heever, and P. Chinowsky, "Planning resilient roads for the future environment and climate change: Quantifying the vulnerability of the primary transport infrastructure system in Mexico," *Transport Policy*, vol. 50, pp. 78–86, 2016.

[9] H. S. Esfahani and M. T. Ramírez, "Institutions, infrastructure, and economic growth," *Journal of Development Economics*, vol. 70, no. 2, pp. 443–477, 2003.

[10] C. César and L. Servén, *The effects of infrastructure development on growth and income distribution*, vol. 270, World Bank Publications, 2004.

[11] J. G. Fernald, "Roads to prosperity? Assessing the link between public capital and productivity," *American Economic Review*, vol. 89, no. 3, pp. 619–638, 1999.

[12] J. E. Neumann, J. Price, P. Chinowsky et al., "Climate change risks to US infrastructure: impacts on roads, bridges, coastal development, and urban drainage," *Climatic Change*, vol. 131, no. 1, pp. 97–109, 2015.

[13] E. Jenelius, T. Petersen, and L.-G. Mattsson, "Importance and exposure in road network vulnerability analysis," *Transportation Research Part A: Policy and Practice*, vol. 40, no. 7, pp. 537–560, 2006.

[14] C. A. Gosse, B. L. Smith, and A. F. Clarens, "Environmentally preferable pavement management systems," *Journal of Infrastructure Systems*, vol. 19, no. 3, pp. 315–325, 2013.

[15] J. Ma, D. Li, L. Cheng, X. Lou, C. Sun, and W. Tang, "Link Restriction: Methods of Testing and Avoiding Braess Paradox in Networks Considering Traffic Demands," *Journal of Transportation Engineering, Part A: Systems*, vol. 144, no. 2, p. 04017076, 2018.

[16] W. Thawat, A. M. Dhareshwar, and P. R. R. Lima, "Vehicle speeds and operating costs," in *Models for Road Planning and Management*, The World Bank, 1987.

[17] National Research Council (US), *Transportation Research Board. Committee for the National Tire Efficiency Study. Tires and Passenger Vehicle Fuel Economy: Informing Consumers, Improving Performance*, vol. 286, 2006.

[18] R. E. Larry et al., "NHTSA tire fuel efficiency consumer information program development: phase 2-effects of tire rolling resistance levels on traction, treadwear, and vehicle fuel economy," in *East Liberty*, National Highway Traffic Safety Administration, Ohio, OH, USA, 2009.

[19] I. Zaabar and K. Chatti, "Calibration of HDM-4 models for estimating the effect of pavement roughness on fuel consumption for U. S. conditions," *Transportation Research Record*, no. 2155, pp. 105–116, 2010.

[20] H. Zhang, G. A. Keoleian, and M. D. Lepech, "Network-level pavement asset management system integrated with life-cycle analysis and life-cycle optimization," *Journal of Infrastructure Systems*, vol. 19, no. 1, pp. 99–107, 2013.

[21] H. Zhang, M. D. Lepech, G. A. Keoleian, S. Qian, and V. C. Li, "Dynamic life-cycle modeling of pavement overlay systems: Capturing the impacts of users, construction, and roadway deterioration," *Journal of Infrastructure Systems*, vol. 16, no. 4, pp. 299–309, 2010.

[22] M. Magnoni, E. Toraldo, F. Giustozzi, and M. Crispino, "Recycling practices for airport pavement construction: Valorisation of on-site materials," *Construction and Building Materials*, vol. 112, pp. 59–68, 2016.

[23] J. Ma, J. Li, and L. Cheng, "Vehicle classification and evaluation method of separating trucks from passenger vehicles," *Shenzhen Daxue Xuebao (Ligong Ban)/Journal of Shenzhen University Science and Engineering*, vol. 32, no. 5, pp. 524–531, 2015.

[24] J. Ma and L. Cheng, "Vehicle Classification and Evaluation Methods for Separating Trucks from Passenger Vehicles," in *Proceedings of the 16th COTA International Conference of Transportation Professionals: Green and Multimodal Transportation and Logistics, CICTP 2016*, pp. 1122–1133, chn, July 2016.

[25] J. Ma and L. Cheng, "The PCA Selection Methods of Reform Plans for Separating Trucks from Passenger Vehicles," in *Proceedings of the 16th COTA International Conference of Transportation Professionals: Green and Multimodal Transportation and Logistics, CICTP 2016*, pp. 1397–1408, chn, July 2016.

[26] S. M. Kamal Hossain, L. Fu, F. Hosseini, M. Muresan, T. Donnelly, and S. Kabir, "Optimum winter road maintenance: Effect of pavement types on snow melting performance of road salts," *Canadian Journal of Civil Engineering*, vol. 43, no. 9, pp. 802–811, 2016.

[27] W. A. Holik, W. H. Schneider, and Q. Huang, "Winter maintenance fleet savings from implementing specialty winter maintenance equipment," *Cold Regions Science and Technology*, vol. 127, pp. 57–64, 2016.

[28] L. J. Leblanc, "An algorithm for the discrete network design problem," *Transportation Science*, vol. 9, no. 3, pp. 183–199, 1975.

[29] R. S. Solanki, J. K. Gorti, and F. Southworth, "Using decomposition in large-scale highway network design with a quasi-optimization heuristic," *Transportation Research Part B: Methodological*, vol. 33, no. 2, pp. 127–140, 1998.

[30] H. Poorzahedy and F. Abulghasemi, "Application of Ant System to network design problem," *Transportation*, vol. 32, no. 3, pp. 251–273, 2005.

[31] B. J. Kim and W. Kim, "An equilibrium network design model with a social cost function for multimodal networks," *Annals of Regional Science*, vol. 40, no. 3, pp. 473–491, 2006.

[32] H. Poorzahedy and M. A. Turnquist, "Approximate algorithms for the discrete network design problem," *Transportation Research Part B: Methodological*, vol. 16, no. 1, pp. 45–55, 1982.

[33] D. Z. W. Wang and H. K. Lo, "Global optimum of the linearized network design problem with equilibrium flows," *Transportation Research Part B: Methodological*, vol. 44, no. 4, pp. 482–492, 2010.

[34] S. Wang, Q. Meng, and H. Yang, "Global optimization methods for the discrete network design problem," *Transportation Research Part B: Methodological*, vol. 50, pp. 42–60, 2013.

[35] Z. Gao, J. Wu, and H. Sun, "Solution algorithm for the bi-level discrete network design problem," *Transportation Research Part B: Methodological*, vol. 39, no. 6, pp. 479–495, 2005.

[36] S.-M. Hosseininasab and S.-N. Shetab-Boushehri, "Integration of selecting and scheduling urban road construction projects as a time-dependent discrete network design problem," *European Journal of Operational Research*, vol. 246, no. 3, pp. 762–771, 2015.

[37] R. Bellman, "On the theory of dynamic programming," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 38, pp. 716–719, 1952.

[38] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1957.

[39] D. Z. W. Wang, H. Liu, W. Y. Szeto, and A. H. F. Chow, "Identification of critical combination of vulnerable links in transportation networks – a global optimisation approach," *Transportmetrica A: Transport Science*, vol. 12, no. 4, pp. 346–365, 2016.

[40] G. M. Schuster and A. K. Katsaggelos, "An optimal polygonal boundary encoding scheme in the rate distortion sense," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 13–26, 1998.

[41] S. Saunders and T. Takaoka, "Solving shortest paths efficiently on nearly acyclic directed graphs," *Theoretical Computer Science*, vol. 370, no. 1-3, pp. 94–109, 2007.