Transportation Research Part C 19 (2011) 708–719

Contents lists available at ScienceDirect

Transportation Research Part C

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

Discretization modeling, integer programming formulations and dynamic programming algorithms for robust traffic signal timing

Jing-Quan Li

California PATH, University of California, Berkeley, Richmond, CA 94804, United States

ARTICLE INFO

Article history:
Received 22 December 2009
Received in revised form 23 December 2010
Accepted 28 December 2010

Keywords:
Robust traffic signal timing
Discretization approach
Integer programming
Dynamic programming

ABSTRACT

Traffic volumes are naturally variable and fluctuate from day to day. Robust optimization approaches have been utilized to address the uncertainty in traffic signal timing optimization. However, due to complicated nonlinear programming models, obtaining a global optimal solution is difficult. Instead of working with nonlinear programming models, we propose a discretization modeling approach, where the cycle, green time, and traffic volume are divided into a finite number of discrete values. The robust signal timing problem is formulated as a binary integer program. Two dynamic programming algorithms are then developed. We obtain optimal solutions for all of the instances with respect to the inputs generated from the discretization.

1. Introduction

Fixed-timed signal control assigns right-of-way at an intersection with pre-determined cycle length, numbers of phases, phase sequences, and individual phase length. While recent studies focus on real-time adaptive signal control, the fixed-timed signal control will still be used for many years due to the costs of detector installation in the adaptive signal control (Smith et al., 2002). In practice, traffic engineers often partition a whole day into several time intervals with relatively stable traffic patterns, such as morning peak, afternoon peak, etc. Certain differential equation techniques (Webster, 1958) and commercial software packages, such as TRANSYT (Vincent et al., 1980), TRANSYT-7F (Wallace et al., 1998) and Synchro (Trafficware, 2003), are then used to investigate timing plans for each time interval. A recent survey on the fixed-timed signal control as well as real-time adaptive control can be found in Papageorgiou et al. (2003).

However, real-world traffic demands are fluctuating in nature, even for the same time interval of day and day of week. Yin (2008) collected traffic data during 9–11AM for an intersection of Gainesville, Florida and showed that the traffic flow varies significantly within that time interval. Therefore, it is desirable to consider uncertain demands in the traffic signal optimization. Based on different approaches to address the issue of uncertain traffic demands, existing studies can be classified into four types.

Type I is to further partition the time interval into certain sub-intervals; and the traffic demands in each sub-interval are assumed to be constant. The traffic signal optimization is conducted over the whole analysis interval, considering the impacts of the former sub-interval on the subsequent sub-interval. The approaches by Han (1996) and Wong et al. (2002) belong to Type I.

Type II is to incorporate the mean, variance and distribution of traffic volumes into the traffic signal optimization. For example, Park et al. (2001) and Park and Kamarajugadda (2007) proposed an integration method to incorporate the mean

E-mail address: jingquan@path.berkeley.edu

0968-090X/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trc.2010.12.009

and variance of traffic demands into the Highway Capacity Manual (HCM) delay equation. The integration method was then combined into a genetic algorithm to investigate timing plans.

Type III is based on different scenarios, where each scenario represents an observation of traffic demands. The occurrence probability is often assigned to each scenario. The deviation of each scenario from the mean value over all the scenarios is often included into the objective function in order to impose penalties for demand changes. Heydecker (1987), Ribeiro (1994), Ukkusuri et al. (2010), and the mean-variance model of Yin (2008) belong to Type III. The conditional value-at-risk model of Yin (2008) is also scenario-based. However, instead of penalizing the demand deviations, the conditional value-at-risk model is used against high-consequence scenarios. Zhang and Yin (2008) then extended the work to synchronize actuated signals robustly on arterial.

Type IV is related to robust optimization notions. Different from the approaches aforementioned, the robust optimization approach does not require accurate demands, demand probability distributions, or a large number of demand scenarios. Only the ranges of uncertain demands are needed. A parameter specified by users can be used to adjust the level of uncertainties. The robust optimization approach often leads to a min–max type of problem. The min–max robust optimization model of Yin (2008) belongs to Type IV.

Type I is suitable to situations where traffic volumes are relatively constant in each smaller sub-interval. Type II is appropriate if the demand probability distribution is known. When the demand distribution is unknown, Type III is more relevant if a large number of scenarios are available and the occurrence probability of each scenario is known. In Type IV, only the range of demands is needed. It is relatively easy to obtain the potential range of the uncertain parameters from historical data. For example, when the traffic volume of movement is varied, it is likely to know the potential range, say 100–200 vehicles/h. Type IV has the least requirements on the knowledge of uncertain demands and may be applicable in most situations. The robust optimization has also been applied to certain problems in the transportation sector, such as vehicle routing (Sungur et al., 2008) and road maintenance (Yin et al., 2008).

Yin (2008) completed a solid study that used a cutting plane algorithm to solve the robust signal optimization. However, as mentioned in Yin (2008), no globally optimal solution is guaranteed to be found with the cutting plane algorithm. The main difficulty is caused by the min–max model itself. Yin and Lawphongpanich (2007) prove that the cutting plan algorithm is convergent if some sufficient conditions hold. Therefore, the robust optimization problem has to be solved multiple times, each time with a different initial solution to obtain a local optimal solution.

In this study, instead of working with continuous nonlinear optimization models, we propose a discretization modeling approach and model the robust signal timing problem using binary integer programs. We show that our approach is able to obtain globally optimal solutions for the min–max model with respect to the discretized inputs.

This paper is organized as follows. Section 2 gives the discretization modeling approach. Section 3 describes the integer programming formulation and dynamic programming (DP) algorithms for the min–max model. Numerical studies are presented in Section 4. Section 5 concludes the paper and describes the future research direction.

2. The robust signal optimization and discretization modeling

Robust optimization techniques require only the potential range of the uncertain data rather than the actual distribution of the uncertain data (Ben-Tal and Nemirovski, 2002). The robust signal optimization model has an outer optimization and an inner optimization problem: the outer optimization provides potential combinations of green times and cycle length; and based on the green times and cycle length from the outer optimization, the inner optimization determines the traffic volumes in uncertain traffic volume set to maximize total traffic delays. The general structure of the robust signal optimization is as follows:

$$\begin{aligned} \min \quad & \max \textit{delays with respect to timings and demands} \\ & \text{subject to}: \textit{uncertain traffic volume set} \\ \text{subject to}: \quad & \textit{certain constraints on green times and cycle length}. \end{aligned}$$

Defining the uncertain traffic volume set is critical to the robust signal optimization. If we conduct the optimization over the worst case scenario, the results may be over-conservative since it is rare that all of the uncertain parameters reach the worst cases simultaneously. Therefore, the robust optimization allows for limits on the potential spaces of the uncertain data. In other words, the robust optimization provide a way to balance feasibility and optimality.

We first provide some definitions. Let N be the set of lane groups. For each lane group $n \in N$, we define M_n as the set of movements. Let $|M| = \sum_{n=1}^{|N|} |M_n|$ be the total number of movements. Fig. 1 provides the signal phasing of NEMA, where movements 2, 4, 6, 8 for the through and right-turn movements, and movements 1, 3, 5 and 7 for the left-turn movements. Assume that the timing plan has four lane groups. Movements 1 and 5 are in lane group 1, movements 2 and 6 are in lane group 2, movements 3 and 7 are in lane group 3, and movements 4 and 8 are in lane group 4. Hence, $N = \{1,2,3,4\}$, $M_1 = \{1,5\}$, $M_2 = \{2,6\}$, $M_3 = \{3,7\}$ and $M_4 = \{4,8\}$. All of the movements in a lane group have the same signal timing; however, the saturation flow and traffic volumes may be different.

Let q_m^{min} and q_m^{max} be the minimum and maximum traffic volume for movement $m \in M_n$, $n \in N$, respectively. Let q_m^0 be the nominal traffic volume of lane group $m \in M_n$, $n \in N$. Yin (2008) defined the uncertain traffic volume set, Q, as

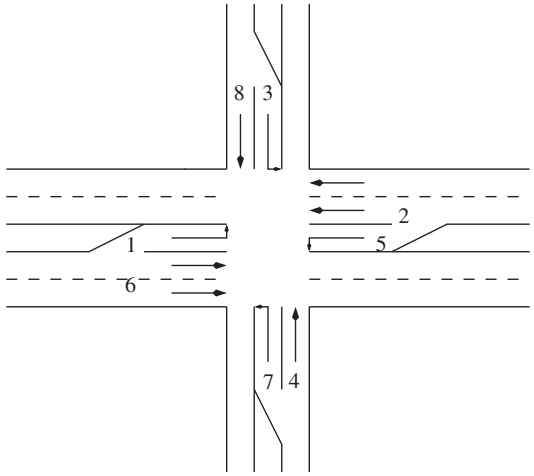


Fig. 1. NEMA signal phasing.

$$Q = \left\{ \left(q_1, q_2, \ldots, q_{\sum_{n=1}^{|N|} |M_n|} \right) \middle| \sum_{n=1}^{|N|} \sum_{m=1}^{|M_n|} \frac{(q_m - q_m^0)^2}{(0.5(q_m^{max} - q_m^{min}))^2} \leqslant \theta^2 \right\}. \quad (1)$$

where q_m^0 is set to $0.5(q_m^{min} + q_m^{max})$, $\sum_{n=1}^{|N|} \sum_{m=1}^{|M_n|} \frac{(q_m - q_m^0)^2}{(0.5(q_m^{max} - q_m^{min}))^2}$ is the total *volume variation*, and θ is a parameter to control the total volume variation. If a traffic engineer favors robustness, a large θ can be used. A typical value of θ is 1.0.

An example is presented to illustrate the aforementioned idea of limiting the uncertain volumes. An intersection consists of two movements (e.g., two one-way roads meet at an intersection). For movement 1, $q_1^{min} = 105, q_1^{max} = 115$, and $q_1^0 = 110$; and for movement 2, $q_2^{min} = 40, q_2^{max} = 60$, and $q_2^0 = 50$. If the worst case scenario is considered, any volume between q_1^{min} and q_1^{max} and any volume between q_2^{min} and q_2^{max} can form a potential combination of volumes. We can think that an infinite θ is used here. If the best case scenario is considered, $\theta = 0$, the volume of movement 1 is 110, and the volume of movement 2 is 50. The best case scenario leads to a deterministic signal optimization problem. The robust optimization approach is between the worst case and best case scenario. For instance, when $\theta = 1.0$, it is infeasible that the volume of movement 1 is 115 and the volume of movement 2 is 60 since the total volume variation is 2.0 in this case. However, it is feasible that the volume of movement 1 is 115 and the volume of movement 2 is 50 since the total volume variation is 1.0.

2.1. Discretization of the delay function

The formula proposed by Webster (1958) is one of the most often quoted models for the delay in the fixed-time traffic control. Certain notable models include Miller (1963), Newell (1965) and McNeil (1968). Dion and Kang (2004) show that the delay equation by the INTEGRATION microscopic simulation generally follows those proposed by Australian Capacity Guide 1981, Canadian Capacity Guide 1995 and HCM 1997. The delay equation in the HCM 2000 is widely used in the signal optimization and is presented as follows:

$$d = \frac{0.5C(1-\lambda)^2}{1 - [\min(1, x)\lambda]} + 900T\left[(x-1) + \sqrt{(x-1)^2 + \frac{4x}{cT}}\right]$$

where d is the average delay per vehicle, C is the cycle length, λ is the effective green split, T is the duration of the analysis period, x is the degree of saturation of the lane group, and c is the capacity of the lane group (Transportation Research Record, 2000).

The HCM delay function is non-convex and non-differentiable. We first use the discretization approach to remove the non-convexity and non-differentiability of the HCM delay function. Let g be the effective green time, q be the traffic volume and s be the saturation flow. Note that $\lambda = g/C$, $x = q/\lambda s = qC/sg$ and $c = \lambda s = gs/C$. The delay formula of HCM 2000 can be rewritten as follows:

$$d(q, C, g) = \frac{0.5C(1 - \frac{g}{C})^2}{1 - \left[\min(1, \frac{qC}{sg})\frac{g}{C}\right]} + 900T\left[\left(\frac{qC}{sg} - 1\right) + \sqrt{\left(\frac{qC}{sg} - 1\right)^2 + \frac{4q^2C}{s^2g^2T}}\right]. \quad (2)$$

The analysis period T and saturation flow s are known. The green time g and cycle length C need to be determined. In general, there is a minimum and a maximum cycle time, say 50 and 140 s. A minimum green time (e.g., 8 s) is often imposed for each

phase for ensuring the fairness in traffic control and safety of pedestrian. If we assume that the minimum time unit is 1, we can list all the potential discrete cycle times ranging from the minimum one to the maximum one, say 50, 51, 52, ..., 140. Similarly, the potential discrete values of green times can be enumerated. If the traffic volume in each lane group is fixed, we can determine the delay for each combination of cycle and green time. Of course, some errors are introduced by the discretization. When the minimum unit is small, the errors between the discrete approximation and continuous delay are small. A similar discretization modeling approach has been used successfully to solve a radar location problem by the author (Mirchandani et al., 2010).

The discretization approach is defined as follows. Let g_n be the effective green time for lane group $n \in N$ and let DC be the sets of possible *discrete* cycle lengths with C_{min} (say, 50 s) and C_{max} (say, 150 s) as the minimum and maximum cycle times, respectively. Let G be the sets of possible *discrete* effective green time with g_{min} as the minimum effective green time. Let L be the cycle lost time. For each $C \in DC$ and $g_n \in G$, $n \in N$, the following equation can be obtained:

$$\sum_{n=1}^{|N|} g_n + L = C.$$

Note that $g_n = C - L - \sum_{i=1, i\neq n}^{|N|} g_i$ and $g_i \geqslant g_{min}$. $C - L - \sum_{i=1, i\neq n}^{|N|} g_i \leqslant C - L - \sum_{i=1, i\neq n}^{|N|} g_{min}$. Therefore, each g_n has a lower bound g_{min} and an upper bound $g_{max} = C - L - \sum_{n=1}^{|N|-1} g_{min}$ for a given cycle time C. Let $A = \{j \in DC \times G | C \in DC, g_{min} \leqslant g_n \leqslant C - L - \sum_{n=1}^{|N|-1} g_{min}\}$ be the set of possible combinations of cycle lengths and green times. Each element $j \in A$ has a unique green time g^j and cycle length c^j.

We now present an example to illustrate the discretization approach. Let $L = 14$, $g_{min} = 8$, $C_{min} = 50$, $C_{max} = 51$ and $|N| = 4$. It is easy to determine $g_{max} = 12$. Hence, if the minimum time unit is 1, the potential cycle lengths include 50 and 51. If the cycle length is 50, the potential green times are 8, 9, 10, 11, and 12. If the cycle length is 51, the potential green times are 8, 9, 10, 11, 12 and 13. Hence, $A = \{(50,8),(50,9),(50,10),(50,11),(50,12),(51,8),(51,9),(51,10),(51,11),(51,12),(51,13)\}$. Table 1 presents the average delay for each potential element in set A when $q = 228$, $s = 1650$ and $T = 0.25$ h.

We propose a formulation for the potential cycle length and green times for all the lane groups. Let y_n^j be a binary variable, with $y_n^j = 1$ if the green time and cycle length of $j \in A$ are assigned to lane group $n \in N$, and $y_n^j = 0$ otherwise. Let C be a variable to represent the cycle length. The set of potential cycle lengths and green times for lane groups, Y, can be identified as follows:

$$Y = \left\{ \left(C, y_1^1, \ldots, y_1^{|A|}, y_2^1, \ldots, y_2^{|A|}, \ldots, y_{|N|}^1, \ldots, y_{|N|}^{|A|}\right) : \begin{array}{lll} \sum_{j\in A} y_n^j = 1 & n \in N & (a1) \\ \sum_{n\in N}\sum_{j\in A} g_n^j y_n^j + L = C & & (a2) \\ \sum_{j\in A} C_n^j y_n^j = C & n \in N & (a3) \end{array} \right\}.$$

Constraints (a1) ensure that each lane group has only one green time/cycle length. Constraints (a2) guarantee that the sum of green times of all the lane groups plus the lost time equals the cycle length. Constraints (a3) force that each lane group has the same cycle length. Note that we use C_n^j and g_n^j in the formulation rather than C^j and g^j to explicitly indicate that the green time and cycle length are associated with a specific lane group. C_n^j simply equals to C_j, and g_n^j simply equals to g_j.

Improta and Cantarella (1984) used a piecewise linear function to approximate the convex delay formula of Webster (1958). Each linear piece consists of a constant, green time multiplied by a coefficient, and cycle length multiplied by another coefficient. Then, the piecewise linear function combined into a binary integer optimization model. However, binary variables in Improta and Cantarella (1984) are to model the precedence relationship of two phases. Our approach is quite different since each binary variable shows if a combination of the green time and cycle length is used in the final solution.

2.2. Discretization of the uncertain demands

Discretizing the delay function alone is not sufficient for the robust signal optimization since the traffic demand of each movement is not constant. As mentioned before, the range of uncertain demands is relatively easy to be obtained.

First, we use the discretization approach for each movement to obtain the discrete traffic volumes. For example, if $q_m^{min} = 105$, $q_m^{max} = 115$, the minimum volume unit is set to 5, we can discretize the continues traffic volumes into the following discrete values: 105, 110 and 115. Let Q_m be the set of indices of possible traffic volumes for movement m, $m \in M_n$, $n \in N$.

Table 1
An example of discrete average delay values (in second): $q = 228$, $s = 1650$ and $T = 0.25$ h.

Cycle	Green					
	8	9	10	11	12	13
50	49.7129	36.7027	29.8434	25.6416	22.7367	
51	53.1863	38.7874	31.2878	26.7654	23.6808	21.3746

Table 2
An example of volume variations.

Volume	105	110	115
Variation	1.00	0.00	1.00

Table 3
An example of average delays with the discrete values of volumes, green times, and cycles.

Cycle	Green	Volume	Average delay	Cycle	Green	Volume	Average delay	Cycle	Green	Volume	Average delay
50	8	105	23.2690	50	8	110	23.6830	50	8	115	24.1192
	9		21.2299		9		21.5320		9		21.8471
	10		19.6121		10		19.8458		10		20.0878
	11		18.2492		11		18.4378		11		18.6320
	12		17.0534		12		17.2104		12		17.3714
51	8	105	24.0252	51	8	110	24.4643	51	8	115	24.9279
	9		21.9146		9		22.2337		9		22.5669
	10		20.2506		10		20.4965		10		20.7515
	11		18.8552		11		19.0532		11		19.2573
	12		17.6351		12		17.7996		12		17.9684
	13		16.5369		13		16.6770		13		16.8202

For each $k \in Q_m$, the corresponding traffic volume is q_m^k. Then for each $C \in DC$, $g_n \in G$ and $k \in Q_m$, we use the discretization approach to obtain the control delay $d(q_m^k, c, g_n)$ for movement m, $m \in M_n$, $n \in N$, as shown in Table 1. The variation of traffic volume $k \in Q_m$ can be determined as

$$v_m^k = \frac{(q_m^k - q_m^0)^2}{(0.5(q_m^{max} - q_m^{min}))^2}. \tag{3}$$

If q_m^k is known, v_m^k can be calculated accordingly. Table 2 gives an example of the volume variation with $q_m^{min} = 105,\ q_m^{max} = 115,\ q_m^0 = (105 + 115)/2 = 110$ and the minimum volume unit 5.

If the green times and cycles presented in Table 1 and potential volumes presented in Table 2 are used, the average delays for potential combinations of green times, cycles, and traffic volumes are presented in Table 3. It is worth mentioning that the discrete values of volumes, green times, and cycles and corresponding delays presented in Table 3 are only for one movement. In the other word, the discretization process for a movement is independent of the discretization process for other movements. After the discretization process are applied to all of movements, an integer programming model is proposed to ensure that the combination of the discrete values of volumes, green times, and cycles for all the movements results into a reasonable signal setting, which will be discussed in Section 3.

2.3. Determine the objective delay function

Commonly used criteria in the traffic signal optimization include minimizing the total number of stops, minimizing the maximum queue length, minimizing the total delays (Webster, 1958) and minimizing the performance index that consists of the delays and stops (Wong et al., 2002). Minimizing the total delays is equivalent to minimizing the average vehicle delay if the traffic volumes are fixed, since the average vehicle delay equals to the total delays divided by the constant total volumes. The average control delay was used to study robust signal timing models in Yin (2008):

$$d = \sum_{n=1}^{|N|} \frac{\frac{0.5C(1-\lambda_n)^2 q_n}{1-[\min(1,x_n)\lambda]} + 900Tq_n[(x_n-1) + \sqrt{(x_n-1)^2 + \frac{4x_n}{cT}}]}{\sum_{n=1}^{|N|} q_n}. \tag{4}$$

where d is the average control delay of all the lane groups, N is the set of lane groups, q_n is the traffic volume for lane group n, λ_n is the effective green split for lane group n, and x_n is the degree of saturation of lane group n.

In the robust traffic signal optimization, traffic volumes are variables. Hence, the average delay cannot be used as an objective function in the discretization approach since otherwise the discretization process for a movement will be dependent on the discretization process for other movements. Therefore, we choose the total traffic delay instead of the average delay. In fact, there is no evidence whether minimizing the total delay is more effective than minimizing the average delay or vice versa.

3. Discrete formulation and dynamic programming algorithms

We now present the discrete formulation for the robust traffic signal optimization problem. Let z_m^k be a binary variable, with $z_m^k = 1$ if traffic volume $q_m^k \in Q_m$ is selected for movement $m \in M_n$, $n \in N$ and $z_m^k = 0$ otherwise. The integer formulation of the min–max robust signal control is as follows:

$$\min_{(C,y_1^1,\ldots,y_{|N|}^{|A|})\in Y} \max \sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} q_m^k d(q_m^k, C, \sum_{j\in A} g_n^j y_n^j) z_m^k \tag{b1}$$

$$\text{st.} \quad \sum_{k\in Q_m} z_m^k = 1 \quad m\in M_n,\ \ n\in N \tag{b2}$$

$$\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} v_m^k z_m^k \leqslant \theta^2. \tag{b3}$$

The objective (b1) of the inner optimization problem is to maximize the total delays, given the green times of each lane group and cycle length. Constraints (b2) ensure only one discrete value of traffic volumes for each lane group. Constraints (b3) force that the total volume variation is within the given range. The outer optimization problem is to minimize the delay from the inner optimization problem.

We claim that the half of the space of traffic volumes for each movement m can be eliminated. Consider the uncertain traffic volume Q in Eq. (1). Assume that there are two possible traffic volumes q_m^L and q_m^R, with $q_m^L < q_m^0 < q_m^R$ and $|q_m^R - q_m^0| = |q_m^0 - q_m^L|$. Observe that q_m^L and q_m^R have the same volume variation (see Eq. (3)). In addition, it is easy to show that $q{\cdot}d(q,C,g)$ is non-decreasing when q is increasing. Therefore, the delay of q_m^R is larger than or equal to the delay of q_m^L, and at the same time q_m^R and q_m^L result in the same volume variations. Since the inner optimization is a max problem, the binary variable corresponding to q_m^R dominates the binary variable corresponding to q_m^L. q_m^L can be eliminated without changing the objective of the min–max problem. All of the traffic volumes less than q_m^0 can be removed, which amounts to the half of the space of total traffic volumes.

The removal of the traffic volumes less than q_m^0 has an interesting implication on the traffic modeling. Without removing the lower half volumes, it is likely that the traffic volume on the through movement is more than its nominal volume, while the traffic volume on the corresponding left movement is lower than its nominal volume. However, it is expected that the traffic demands on these two movements increase or decrease at the same time since two movements are relevant. The removal of the lower half volume guarantees that the volumes of both through and left movements increase simultaneously, which is more expected in the traffic modeling.

Observe the constraints (b2) where each movement can independently select a potential traffic volume. The maximum number of volume combinations among the movements are $\prod_{m\in M_n, n\in N}|Q_m|$. Therefore, the inner problem is pseudo-polynomially solvable. The outer optimization problem is not known to belong to NP since the objective of the outer problem is related to the inner optimization problem (Smith et al., 2009).

3.1. A dynamic programming algorithm

We design a dynamic programming algorithm to solve the min–max problem. Dynamic programming has been used in the traffic signal optimization (Cai et al., 2009).

Note that the cycle length C is a variable in the outer optimization problem. We first break down the min–max problem into $|DC|$ min–max problems, each of which has a fixed cycle length $C \in DC$. For each of $|DC|$ min–max problems, we solve it by using the dynamic programming. The optimal solution can be found from the solutions of $|DC|$ min–max problems.

Each state is represented by a label, (R,T,V,i), where $i \in N$ is the last reached lane group, V is the total volume variations, and T is the total green times. An additional resource vector, R, is included to indicate the green times for the lane groups that have been visited by this label. The delay of label (R,T,V,i) is $d(R,T,V,i)$, representing the total delays among all the visited lane groups. Let the lane The first label is $(\emptyset,0,0,0)$ with delay 0.

Now consider that label (R,T,V,i) is extended to a label in the next lane group, $(R',T',V',i+1)$, when i is not the last lane group. Label $(R',G',V',i+1)$ is determined as follows. First, a green time $g \in G$ needs to be selected. $T' = T + g$. g is a feasible green time if (i) $T' + L = C$ when $i+1$ is the last lane group; and (ii) $T' + \sum_{i+2}^{|N|} g_{min} + L \leqslant C$ when $i+1$ is not the last lane group. $R'_{i+1} = g$ and $R'_k = R_k$ for $k \leqslant i$. If g is a feasible green time for lane group $i+1$, traffic volume q_m^k for each movement $m \in M_{i+1}$ of lane group $i+1$ needs to be selected. Note that we need to consider all of the combinations of traffic volumes for each movement of lane group $i+1$. The corresponding volume variation $\sum_{m\in M_{i+1}} v_m^k$ is determined using Eq. (3). $V' = V + \sum_{m\in M_{i+1}} v_m^k$. The extension is feasible if $V' \leqslant \theta^2$.

Label dominance is extremely important to design an efficient dynamic programming algorithm. Assume in the case of extensions that both label (R,T,V,i) and label $(\overline{R},\overline{T},\overline{V},i)$ have lane group i as their last reached lane group. First, observe that if a label dominates another label, two labels need to have the exactly same green times for each lane group that has been visited. This observation follows our min–max formulation. When i is not the last lane group, label (R,T,V,i) dominates label $(\overline{R},\overline{T},\overline{V},i)$ if (i) $R = \overline{R}$, (ii) $d(R,T,V,i) \geqslant d(\overline{R},\overline{T},\overline{V},i)$, and (iii) $V \leqslant \overline{V}$. If i is the last lane group, we do not need to consider the total volume variations since no extension is needed. Label (R,T,V,i) dominates label $(\overline{R},\overline{T},\overline{V},i)$ if (i) $R = \overline{R}$ and (ii) $d(R,T,V,i) \geqslant d(\overline{R},\overline{T},\overline{V},i)$. Note that there is only one label in the last lane group for each potential timing plan R.

After all extensions are finished, we can examine the label in the last lane group and find the best one with the minimal total delay, which corresponds to the outer minimization problem. The overall algorithm with cycle length $C \in DC$ is outlined as follows.

Step 1 (Initialization):Place the first label into the active label pool.
Step 2 (Inner maximization):For all the labels (R,T,V,i) in the active label pool:

Step 2.1: Remove the current label from the active label pool.
Step 2.2: For all the feasible green times of lane group $i+1$
Step 2.2.1: For all the possible volume combinations of all the movements of lane group $i+1$
Step 2.2.1.1: Determine if a new label can be extended from the current label given the total volume variation.
Step 2.2.1.2: If the new label is generated, not dominated and $i+1 \neq |N|$, insert it into the active label pool. Assess whether the new label can dominate other existing labels of the corresponding lane group $i+1$ and timing plan R. If so, delete these dominated labels.

Step 3 (Outer minimization):Examine the labels in the last lane group and output the label with the minimal delay as the optimal one.

3.2. An enhanced bi-directional dynamic programming algorithm

As mentioned above, reducing the number of non-dominated labels is very important to the DP algorithm. We conducted some preliminary experiments for an intersection with four lane groups and cycle length 50, and the numbers of non-dominated labels for four lane groups are: 685, 10,716, 119,183, and 35, respectively. Lane group 4 has the minimum number of non-dominated labels, while lane group 3 has the largest number of non-dominated labels.

We also have the following observations: (1) the number of non-dominated labels for lane group 1 is small since the combinations are caused by the traffic volumes of the movements of lane group 1 and the potential green time; (2) the number of non-dominated labels for lane group 2 is larger than the number of non-dominated labels for lane group 1 since the combinations are caused by two lane groups; (3) the number of non-dominated labels for lane group 3 significantly exceeds the number of non-dominated labels for lane group 2, since three lane groups are involved; and (4) the number of non-dominated labels in lane group 4 is very small, since each potential timing plan only needs to keep one non-dominated label.

Therefore, the performance of the DP algorithm can be substantially improved if the number of non-dominated labels for the third lane group is reduced in some way. We designed a bi-directional dynamic programming algorithm to achieve this goal. First, we started the DP from lane group 1, extended to lane group 2, and stop. Second, we started the DP from lane group 4, extended to lane group 3, and stop. Finally, we attempted to join the labels of lane groups 2 and 3 together by examining if (1) the cycle length is exactly matched and (2) the total volume variation is no more than θ^2. The bi-direction DP has been successfully used to solve the elementary resource constrained shortest path problem (Righini and Salani, 2006) and traveling salesman problem with time windows (Li, 2009).

3.3. Benefits of the discretization and discrete formulation

Besides the ability of handling the non-convexity and non-differentiability of the HCM delay function, the use of the discretization approach has following benefits.

First, the signal timing generated by continuous models is likely to be fractional. In order to implement the timing in signal controllers, traffic engineers have to round the fractional values to nearest integers. It is not clear how much inaccuracy the rounding will yield. In contrast, the timing by the discretization approach is directly integral; and the rounding is not needed.

Second, consider that the Webster's formula with first two terms is used in the objective function. Gallivan (1982) proves that Webster's formula with first two terms is convex in the variables of $1/c$ and g/c, where c is the cycle length and g is the green time. Minimizing such convex function is a simple problem and can be solved by the interior point method. However, the robust signal optimization is a min–max model. Variables in the inner model are now traffic demands (q). The inner model is to maximize a highly nonlinear function in the variables of q, which is difficult to obtain the optimal solution even for the inner optimization model itself. The overall problem is much more complicated after the outer optimization model is considered. Therefore, even using a simpler delay function results in a difficult optimization problem. In contrast, no matter

Table 4
Min, max, and saturation flows for two examples.

Group	Example 1					Example 2		
		Under-saturated		Over-saturated				
	Saturation flow	Min flow	Max flow	Min flow	Max flow	Saturation flow	Min flow	Max flow
1	1900	100	350	100	450	1650	168	288
2	3800	200	600	250	800	3200	780	1348
3	3800	400	900	550	1200	1650	188	408
4	1900	150	400	150	400	1700	88	208
5	1900	200	300	200	500	1650	28	100
6	3800	300	700	300	1000	3200	860	1252
7	3800	500	800	600	1200	1650	32	92
8	1900	120	220	120	380	1700	296	656

Table 5
Efficiency of the bi-directional DP based on example 2 with $\theta = 0.6$.

Cycle (s)	50	51	52	53	54	55	56	57	58	59	60
Obj	875,323	802,204	742,831	695,189	655,374	612,721	578,481	546,645	516,070	493,225	466,171
DP CPU (s)	5.27	7.27	9.55	11.74	15.18	19.41	26.32	30.53	40.91	44.51	53.52
DP label	26,307	40,779	59,862	83,757	113,350	149,131	191,540	241,514	299,368	366,045	441,922
Bi-DP CPU (s)	2.94	3.17	3.38	3.84	4.17	4.62	5.26	5.88	7.00	7.57	8.74
Bi-DP label	8607	11,968	15,839	20,259	25,267	30,868	36,969	43,654	50,852	58,481	66,782

Table 6
Certain near optimal results of the min–max problem for example 2 with $\theta = 0.5$.

Cycle (s)	Green times (s)				Delay (s)	Average delay	Delay (s) with unit vol	Percentage (%) (s)
	Group 1	Group 2	Group 3	Group 4				
97	12	36	27	8	241,824	67.7948	242,281	99.81
98	12	36	28	8	241,821	67.6044	242,282	99.81
99	**12**	**37**	**28**	**8**	**241,237**	**67.5356**	**241,660**	**99.83**
100	12	37	29	8	241,681	67.5653	241,911	99.90
101	12	38	29	8	241,511	67.6123	241,762	99.90
102	13	38	29	8	241,286	67.6440	241,684	99.84
103	13	39	29	8	241,805	67.9992	242,209	99.83
104	13	39	30	8	241,448	67.5946	241,692	99.90
105	13	39	30	9	241,793	67.7861	242,164	99.85
106	13	40	30	9	241,940	68.0371	242,254	99.87
107	13	40	31	9	241,653	67.6520	241,994	99.86

whether a simple or complicated delay function is used, our discretization and integer programming approach can obtain the optimal solution with respect to the discretized inputs at expense of relatively long computational time.

4. Numerical studies

Computational experiments were conducted to examine our discretization approach and the corresponding algorithms for the min–max model. Yin (2008) proposed two examples with the following parameters: the minimum cycle length (C_{min}) is 50 s, the maximum cycle length (C_{max}) is 140 s, the lost time (L) is 14 s, the minimum green time (g_{min}) is 8 s, and the analysis period (T) is 0.25 h. Example 1 has four lane groups: movements 1 and 6 belong to lane group 1; movements 2 and 5 belong to lane group 2; movements 3 and 8 belong to lane group 3; and movements 4 and 7 belong to lane group 4. Example 2 is from a real-world intersection in the City of Lynnwood, Washington. There are also four lane groups: lane group 1 includes movements 1 and 5; lane group 2 includes movements 2 and 6; and lane groups 3 and 4 are the same as in example 1. There are 36 scenarios with different traffic volumes in example 2; and each scenario has the same occurrence probability. The min–max model is examined for both example 1 and example 2. The minimum, maximum and saturation flows for examples 1 and 2 were extracted from Tables 1 and 7 of Yin (2008) and are presented in Table 4.

Table 7
Certain near optimal results of the min–max problem for example 1 with under-saturated flows.

θ	Cycle	Green times (s)				Delay (s)	Average delay	Delay (s) unit vol with	Percentage (%) (s)
		Group 1	Group 2	Group 3	Group 4				
0.5	57	10	9	12	12	115,789	35.0345	116,198	99.65
	58	**10**	**9**	**13**	**12**	**114,196**	**34.2417**	**114,453**	**99.78**
	59	10	9	13	13	114,904	34.4540	115,065	99.86
	60	10	10	13	13	115,327	34.6848	115,617	99.75
	61	11	9	14	13	115,847	34.7889	116,055	99.82
	62	11	10	14	13	114,485	34.3283	114,714	99.80
	63	11	10	14	14	115,034	34.4930	115,211	99.85
1.0	64	12	10	15	13	139,752	39.8722	140,097	99.75
	65	12	10	15	14	141,841	41.0538	142,183	99.76
	66	12	10	16	14	140,445	40.5325	140,686	99.83
	67	12	11	16	14	139,366	40.2211	139,599	99.83
	68	13	11	16	14	140,590	40.4576	140,964	99.73
	69	13	11	16	15	142,834	41.3412	143,157	99.77
	70	**13**	**11**	**17**	**15**	**137,764**	**38.6978**	**137,832**	**99.95**
	71	13	12	17	15	139,069	39.4522	139,239	99.88

The following minimum volume units are used for example 2: for movements 5 and 7, we set the minimum volume unit as 1; and for other movements, we set the minimum volume unit as 5. For example 1 with under-saturated flows, we use 5 as the minimum sampling units for movements 5 and 8 and 10 for other movements. We used 10 as the sampling units for all of the movements in the case of over-saturated flows.

All of the algorithms were implemented in C++ on a Linux workstation with eight Intel processors at 2 GHz, 16 GB of RAM and a 32-bit Linux operating system. It should be mentioned that all the algorithms are sequential and do not take advantage of the availability of multiple processors.

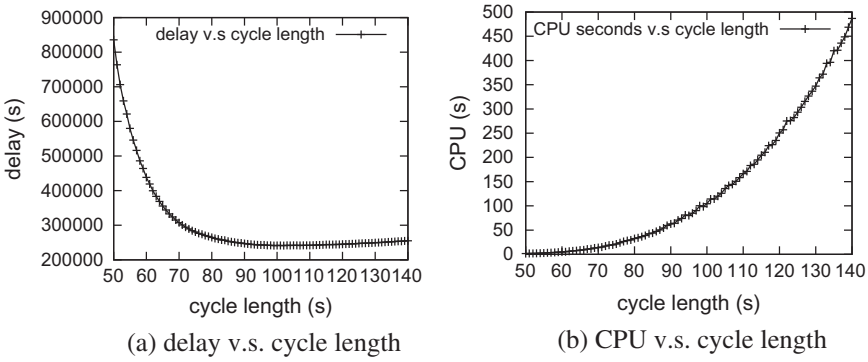
4.1. Efficiency of the bi-directional DP

We first examined the efficiency of the bi-directional DP. Table 5 presents the computational results for example 2 with $\theta = 0.6$ and cycle lengths from 50 s to 60 s. The first row is the cycle length; the second row is the delay; the third and fourth rows are CPU seconds and the number of non-dominated labels, respectively, when the single directional DP is used. The fifth and sixth rows are CPU seconds, and the number of non-dominated labels if the bi-directional DP (Bi-DP) is used. As we expect, the traditional DP and bi-directional DP achieve the same delay. However, the computational time of the bi-directional DP is significantly smaller than the time of the single directional DP.

4.2. Numerical results for examples 1 and 2

Fig. 2 gives the impacts of the cycle length on delays and CPU seconds based on example 2. The longer cycle length requires much longer computational time since the longer cycle yields more potential timing plans. When the cycle length is small, the delay is very large. The delay is decreased significantly when the cycle length is increased. When the delay reaches the minimum, it will slightly increase if the cycle length continues to increase, as seen in Fig. 2(a).

Tables 6–8 present certain near optimal results for example 2, example 1 with under-saturated flows, and example 1 with over-saturated flows, respectively. In Table 6, column 1 is the cycle length; columns 2 through 5 give the green times for each lane group; column 6 is the total delay; and column 7 is the average delay, which equals to the total delay divided by the



(a) delay v.s. cycle length (b) CPU v.s. cycle length

Fig. 2. Delays and CPU seconds with different cycle lengths.

Table 8
Certain near optimal results of the min–max problem for example 1 with over-saturated flows.

θ	Cycle	Green times (s)				Delay (s)	Average delay	Delay (s) with unit vol	Percentage (%) (s)
		Group 1	Group 2	Group 3	Group 4				
0.5	102	20	18	25	25	319,807	72.5186	320,500	99.78
	103	20	18	25	26	319,559	72.4624	319,893	99.90
	104	**20**	**18**	**26**	**26**	**318,813**	**72.2932**	**319,422**	**99.81**
	105	20	19	26	26	320,105	72.7511	320,905	99.75
	106	21	18	26	27	320,552	72.6875	320,832	99.91
	107	21	18	27	27	319,849	72.5281	320,602	99.77
	108	21	19	27	27	319,090	72.3560	319,865	99.76
1.0	116	24	20	29	29	450,354	95.6166	450,933	99.87
	118	**24**	**20**	**30**	**30**	**448,911**	**94.9072**	**449,421**	**99.89**
	119	25	20	30	30	451,368	96.6527	452,198	99.82
	120	25	20	31	30	450,306	95.6064	450,594	99.94
	121	25	20	31	31	449,010	94.9281	449,453	99.90

corresponding total volumes. The optimal solutions are highlighted in bold. Tables 7 and 8 have similar column definitions with Table 6 except that column 1 represents θ. Column 7 in Table 6 and columns 8 in Tables 7 and 8 show that when a signal timing yields the minimal total delays, the corresponding average delay is also the smallest among all of the average delays with other timings.

In order to investigate the effects of volume sampling units, we (1) use the signal timings from columns 2 to 6, (2) set the minimum volume unit as 1 for all of the movements and (3) solved the problem to obtain the delay. The result is presented in column 8 of Table 6 and columns 9 in Tables 7 and 8. Column 9 of Table 6 is the percentage of column 6 over column 8. Column 10 of Tables 7 and 8 is the percentage of column 7 over column 9. The total delays that are obtained with the same timings and different volume sampling units are very close, as can be seen in the last column. Therefore, the sampling volume units used in our experiments are very satisfactory.

4.3. Rough comparison with the cutting plane algorithm by Yin (2008)

Different objective functions are used in the inner optimization problem: the cutting plane algorithm by Yin (2008) is to maximize the average delay, while our DP approach is to maximize the total delay. In order to give comparisons between the DP algorithm and the cutting plane algorithm, we first apply the DP algorithm to obtain the best timing plan with the objective of minimizing the maximal total delay. Then, we obtain the best timing plan from Yin (2008). Finally, based on two timing plans, we solve the inner optimization problem, which attempts to maximize the average delay with the respect of discrete inputs.

Let g_n be the green time for lane group $n \in N$. Let z_m^k be a binary variable, with $z_m^k = 1$ if traffic volume $q_m^k \in Q_m$ is selected for movement $m \in M_n$, $n \in N$ and $z_m^k = 0$ otherwise. The formulation of maximizing the average delay is as follows:

$$\max \frac{\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} q_m^k d(q_m^k, C, g_n) z_m^k}{\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} q_m^k z_m^k} \tag{c1}$$

$$\text{st} \quad \sum_{k\in Q_m} z_m^k = 1 \quad m \in M_n, \ \ n \in N \tag{c2}$$

$$\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} v_m^k z_m^k \leqslant \theta^2 \tag{c3}$$

where the objective (c1) is to maximize the average delay, and constraints (c2) and (c3) are as same as constraints (b2) and (b3) respectively. The formulation results into a nonlinear integer programming problem due to the nonlinear objective function. The objective function can be linearized as follows. First, introduce a new variable t and let $t = \frac{\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} q_m^k d(q_m^k, C, g_n) z_m^k}{\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} q_m^k z_m^k}$. Second, add a constraint $\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} q_m^k d(q_m^k, C, g_n) z_m^k \geqslant t(\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} q_m^k z_m^k)$. Finally, linearize tz_m^k using the big M method. The resulting linear integer programming model is as follows:

$$\max t \tag{d1}$$

$$\text{st} \sum_{k\in Q_m} z_m^k = 1 \quad m \in M_n, \ \ n \in N \tag{d2}$$

$$\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} v_m^k z_m^k \leqslant \theta^2 \tag{d3}$$

$$\sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} q_m^k d(q_m^k, C, g_n) z_m^k \geqslant \sum_{n\in N}\sum_{m\in M_n}\sum_{k\in Q_m} q_m^k x_m^k \tag{d4}$$

$$x_m^k \leqslant t \quad k \in Q_m, \ \ m \in M_n, \ \ n \in N \tag{d5}$$

$$x_m^k \leqslant M z_m^k \quad k \in Q_m, \ \ m \in M_n, \ \ n \in N \tag{d6}$$

$$x_m^k \geqslant t + M(z_m^k - 1) \quad k \in Q_m, \ \ m \in M_n, \ \ n \in N \tag{d7}$$

where x_m^k is a variable to linearize tx_m^k (constraints d5, d6 and d7), and M is a big number. The mixed-linear integer programming model is solved using the CPLEX.

Table 9 presents the maximal average delay when the signal timings are obtained from the DP approach and cutting plane algorithm respectively. The DP approach decreases the maximal average delays as well (see column 9 in Table 9).

However, we have to state that the comparisons here are just rough since two algorithms have different types of inputs: the inputs of the DP are discretized, while the inputs of the cutting plane (Yin, 2008) are continuous. As mentioned before, certain sampling errors are introduced in the discretization process. Thus, it is unfair to say that the DP algorithm obtains better results than the cutting plane algorithm.

4.4. Computational time

Table 10 presents the delay, cycle length and CPU minutes with different θ. It is clear that the delay and cycle length are increased when θ is increased, which is because a larger θ leads to a larger set of feasible traffic volumes. The inner maxi-

Table 9
Comparison with the cutting plan algorithm by Yin (2008).

Instance	θ	Study	Cycle	Green times (s)				Average delay (s)
				Group 1	Group 2	Group 3	Group 4	
Example 2	0.5	Yin (2008)	100	12	39	26	9	74.6980
		Ours	99	12	37	28	8	72.6046
Example 1, under-saturation	0.5	Yin (2008)	59 (58)	10	9	13	12	36.6475
		Ours	58	10	9	13	12	36.6475
	1.0	Yin (2008)	68	13	11	16	14	51.0325
		Ours	70	13	11	17	15	43.1871
Example 1, over-saturation	0.5	Yin (2008)	102	20	18	25	25	75.5836
		Ours	104	20	18	26	26	75.5649
	1.0	Yin (2008)	116 (115)	24	19	29	29	99.5448
		Ours	118	24	20	30	30	98.4879

Table 10
Computational times of different instances.

	Example 2	Example 1, under-saturated		Example 1, over-saturated	
θ	0.5	0.5	1.0	0.5	1.0
Delay	241,237	114,196	137,764	318,813	448,911
Cycle	99	58	70	104	118
CPU (min)	206	24	217	77	761

mization problem will produce a larger delay. When θ increases, the computational time increases significantly. For example 1 with over-saturated flows, 761 CPU minutes (around 12.7 h) are needed to solve the problem to optimality if θ equals 1. The larger traffic volume ranges result in longer computational time. For instance, example 1 with over-saturated flows requires much larger computational time of corresponding instances with under-saturated flows. When the traffic volume or θ is reduced, the computational time is significantly reduced. For instance, only 24 CPU minutes are needed for the example 1 with under-saturated flows and $\theta = 0.5$. As mentioned before, the DP algorithm has pseudo-polynomial complexity, which depends on the traffic volume ranges. Our DP algorithm requires much longer computational time than the nonlinear programming approach in Yin (2008). Since the optimization for fixed-time signalized intersections is done offline, the computational time in such range is acceptable.

4.5. Benefits of the robust signal optimization

In this paper, we do not focus on investigating how the robust timing performs better than non-robust versions. In order to show the benefits of the robust signal optimization, we cite some results from Yin (2008), where Monte-Carlo macroscopic simulations were used to study the performances of different methods with uncertain traffic volumes. The simulation conducted on example 2 shows that, in comparison with results from the non-robust algorithm, (1) the worst-case delay is dropped by 4.9%; and (2) the standard deviation of delays per vehicle is dropped by 12.0% (see Table 9 in Yin (2008)). Therefore, it is clear that the timing plans generated by the robust signal optimization are less sensitive to the variations in traffic flows. Interested readers are referred to Yin (2008) for more details.

5. Conclusion and future work

In this paper, we considered the traffic signal optimization problem with uncertain traffic volumes. Thanks to relatively wide applicability, the robust optimization based model proposed by Yin (2008) was selected to address the uncertain volumes. However, due to the complexity of the nonlinear programming model (Yin, 2008; Yin and Lawphongpanich, 2007), no globally optimal solution was guaranteed to be found for these models.

Instead of working with nonlinear programming models, we proposed a discretization modeling approach, where the cycle, green time, and traffic volume were divided into a finite number of discrete values. When the minimum sampling unit is small, the errors between the discrete approximation and the continuous delay are small. We formulated the robust signal timing problem as a binary integer program. A dynamic programming algorithm was designed for solving the min–max problem. We then proposed an enhanced bi-directional dynamic programming to reduce the computational time.

We obtained the global optimal solutions for all the instances with respect to the inputs generated from the discretization. Our approach produced comparable solutions to the continuous model of Yin (2008). The solutions are improved for many instances. The total computational time could be as long as 12 h if the traffic volume range and traffic variation are

large, while 20 min are sufficient to finish the computation when the volume range or variation is small. The computational time in such range is acceptable since the computation is conducted offline.

As seen in this paper, the dynamic programming suffered from input data with a large range. Future research may investigate the cutting plane technique (Smith et al., 2009) to avoid this problem. The state space relaxation technique (Christofides et al., 1981) may also be used to reduce the computational time in the dynamic programming.

Acknowledgments

The author would express his sincere thanks to Guoyuan Wu for helpful discussions. The author would also express his sincere thanks to the anonymous referees for their valuable suggestions.

References

Ben-Tal, A., Nemirovski, A., 2002. Robust optimization-methodology and applications. Mathematical Programming 92 (3), 453–480.
Cai, C., Wong, C.K., Heydecker, B.G., 2009. Adaptive traffic signal control using approximate dynamic programming. Transportation Research Part C – Emerging Technologies 17 (5), 456–474.
Christofides, N., Mingozzi, A., Toth, P., 1981. State-space relaxation procedures for the computation of bounds to routing problems. Networks 11 (2), 145–164.
Dion, F.H.H.R., Kang, Y.-S., 2004. Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections. Transportation Research Part B – Methodological 38 (2), 99–122.
Gallivan, S., 1982. A geometric proof that webster's two term formula for a delay at a traffic signal is convex. Tech. rep., Transport Studies Group, University College of London.
Han, B., 1996. Optimising traffic signal settings for periods of time-varying demand. Transportation Research Part A – Policy and Practice 30 (3), 207–230.
Heydecker, B., 1987. Uncertainty and variability in traffic signal calculations. Transportation Research Part B – Methodological 21 (1), 79–85.
Improta, G., Cantarella, G.E., 1984. Control system design for an individual signalized junction. Transportation Research Part B – Methodological 18 (2), 147–167.
Li, J.-Q., 2009. A computational study of bi-directional dynamic programming for the traveling salesman problem with time windows. Working paper.
McNeil, D.R., 1968. A solution to the fixed-cycle traffic light problem for compound poisson arrivals. Journal of Applied Probability 5 (3), 624–635.
Miller, A.J., 1963. Settings for fixed-cycle traffic signals. Operations Research Quarterly 14 (4), 373–386.
Mirchandani, P.B., Li, J.-Q., Long, Y., 2010. Locating a surveillance infrastructure in and near ports or on other planar surfaces to monitor flows. Computer-Aided Civil and Infrastructure Engineering 25 (2), 89–100.
Newell, G.F., 1965. Approximation methods for queues with application to the fixed-cycle traffic light. SIAM Review 7 (2), 223–240.
Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., Wang, Y., 2003. Review of road traffic control strategies. Proceedings of the IEEE 91 (12), 2043–2067.
Park, B.B., Kamarajugadda, A., 2007. Development and evaluation of a stochastic traffic signal optimization method. International Journal of Sustainable Transportation 1 (3), 193–207.
Park, B.B., Rouphail, N.M., Sacks, J., 2001. Assessment of stochastic signal optimization method using microsimulation. Transportation Research Record 1748, 40–45.
Ribeiro, P.C.M., 1994. Handling traffic fluctuation with fixed-time plans calculated by TRANSYT. Traffic Engineering & Control 35 (6), 362–366.
Righini, G., Salani, M., 2006. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optimization 3 (3), 255–273.
Smith, B.L., Scherer, W.T., Hauser, T.A., Park, B.B., 2002. Data-driven methodology for signal timing plan development: a computational approach. Computer-Aided Civil and Infrastructure Engineering 17 (6), 387–395.
Smith, J.C., Lim, C., Alptekinoğlu, A., 2009. New product introduction against a predator: a bilevel mixed-integer programming approach. Naval Research Logistics 56 (8), 714–729.
Sungur, I., Ordónez, F., Dessouky, M., 2008. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. IIE Transactions 40 (5), 509–523.
Trafficware, 2003. Synchro 6 User's Guide. Trafficware, Albany, CA.
Transportation Research Record, 2000. Highway capacity manual. Tech. rep., Transportation Research Record.
Ukkusuri, S.V., Ramadurai, G., Patil, G., 2010. A robust transportation signal control problem accounting for traffic dynamics. Computers and Operations Research 37 (5), 869–879.
Vincent, R.A., Mitchell, A.T., Roberson, D.T., 1980. User guide to TRANSYT version 8. Tech. Rep. TRRL Report LR888, Transport and Road Research Laboratory, Crowthorne.
Wallace, C.E., Courage, K.G., Hadi, M.A., Gan, A.G., 1998. TRANSYT-7F User's Guide. University of Florida, Gainesville, FL.
Webster, F.V., 1958. Traffic signal settings. Tech. Rep. Road Research Technical Paper No. 39, Her Majestry's Stationery Office.
Wong, W.T., Wong, S.C., Leung, C.M., Tong, C.O., 2002. Group-based optimization of a time-dependent TRANSYT traffic model for area traffic control. Transportation Research Part B – Methodological 36 (4), 291–312.
Yin, Y., 2008. Robust optimal traffic signal timing. Transportation Research Part B – Methodological 42 (10), 911–924.
Yin, Y., Lawphongpanich, S., 2007. A robust approach to continuous network designs with demand uncertainty. In: Allsop, R.E., Bell, M.G.H., Heydecker, B.G. (Eds.), Transportation and Traffic Theory 2007. Elsevier, pp. 111–126.
Yin, Y., Lawphongpanich, S., Lou, Y., 2008. Estimating investment requirement for maintaining and improving highway systems. Transportation Research Part C – Emerging Technologies 16 (2), 199–211.
Zhang, L., Yin, Y., 2008. Robust synchronization of actuated signals on arterials. Transportation Research Record 2080, 111–119.