The Model and Properties of the Traffic Light Problem *

Chang-Biau Yang and Yun-Jaw Yeh
Department of Applied Mathematics
National Sun Yat-sen University
Kaohsiung, Taiwan 80424
cbyang@math.nsysu.edu.tw

Abstract

The traffic network of a city can be modeled by a graph in which vertices represent the intersections of roads and edges represent the segments of roads. The traffic light problem is concerned with the setting of the starting time of the green light for each traffic light such that the total waiting time of the traffic in the traffic network is minimized. In this paper, we first prove that the traffic light problem is NP-hard for the traffic light cycle $T \geq 3$, that the traffic light problem on a planar graph with $T = 2$ can be solved by the algorithm designed for the weighted maximum cut problem, whose time complexity is $O(n^{\frac{3}{2}} \log n)$, and that the two-phase setting problem (a special case of the traffic light problem) on a planar graph can also be solved in the same time complexity. Then, we discuss some properties of the traffic light problem. Using these properties, we design an algorithm to find the minimum penalty of a polygon with n edges in $O(n)$ time. Finally, we present a heuristic algorithm to solve the traffic light problem and show some experiment results. **Keywords:** traffic light, graph, NP-hardness, maximum cut.

1 Introduction

The traffic condition in a city is greatly affected by the timing setting of the traffic lights, i.e., when the red light and the green light is on. In this paper, we consider the following problem: decide the starting time of the green light for each traffic light in such a way that the total waiting time of the traffic in the traffic network is minimized.

The traffic light problem has been studied extensively under various assumptions. Some researchers [2, 3, 5–7] used sensors to get the flow of the traffic and used many kinds of methods to decide the timings of the traffic lights. The study on fuzzy systems has also been applied to the traffic light problem [6, 7]. For the simpler conditions of roads, J. Favilla *et al.* [3] put a fuzzy logic controller (FLC) into the traffic controller.

Because the general traffic light problem is of practical interest but difficult to solve, we will make some assumptions to simplify the problem: (1) Each traffic light has only two lights, green and red. (2) All traffic lights have the same cycle (period), T, i.e., T equals to the duration of one green light and one red light. (3) The green light duration is equal to the red light duration for all the traffic lights. If a car reaches an intersection and the traffic light in its direction is red, it must wait until the light turns green, and then starts to drive toward the next intersection. (4) Each car takes one unit of time to pass through one unit of distance.

We can use a graph $G = (V, E)$ to represent the traffic network of a city, where each vertex in V corresponds to an intersection of some roads, and each edge (u, v) in E corresponds to the segment of a road connecting the two intersections u and v. Each edge (u, v) is associated with a nonnegative integral length (distance) $d(u, v)$. For the simplified version of the traffic light problem, we assume that there is only one traffic light at each intersection (vertex). We can use the starting time of a green light, denoted as $t(v)$, to specify the timing setting of the traffic light at vertex v, i.e., the traffic light at v is green in the time intervals $[t(v)+q\cdot T, t(v)+(q+1/2)\cdot T)$, for all integer q, and red in all other time intervals. For the realistic version, we assume that there is one traffic light for each edge (u, v) incident to a vertex v. Therefore, we use $t(u, v)$ to denote the starting time of a green light of the traffic light corresponding to edge (u, v) incident to v. The traffic light problem is then to assign $t(v)$ for each $v \in V$ such that this kind of waiting time can be minimized.

The rest of the paper is organized as follows. In Section 2, we first give the definition of the penalty of an edge and then give the formal definition of the traffic light problem, including the simplified version and the

*This research work was partially supported by the National Science Council of the Republic of China under contract NSC 85-2213-E-110-011.

realistic version. In Section 3, we prove that the traffic light problem is NP-hard for the traffic light cycle $T \geq 3$. Then, we show how to solve the problem on a planar graph with $T = 2$ by the algorithm designed for the weighted maximum cut problem [8], which has a time complexity $O(n^{\frac{3}{2}} \log n)$. In Section 4, we first define the two-phase setting problem, and solve it by using the weighted maximum cut algorithm if the underlying traffic network is a planar graph. Then we discuss some properties of the traffic light problem. In Section 5, we first give an algorithm to find the minimum penalty of a polygon with n edges in $O(n)$ time. We then use the result to design a heuristic algorithm for solving the traffic light problem. In Section 6, we give some experiment results. Finally, the paper concludes with Section 7. Note that all proofs of lemmas and theorems, and some examples are removed due to the space limitation of the conference.

2 The Definition of the Traffic Light Problem

A traffic network is represented by a graph $G = (V, E)$, where each edge (u, v) in E corresponds to a segment of a road and each vertex v in V corresponds to the intersection of some roads. Note that we assume that each road (segment) is bi-directional. Each edge (u, v) is also associated with a nonnegative integral distance, $d(u, v)$, and each vertex has some traffic lights (see Figure 1). In the *simplified traffic light* problem, there is only one traffic light on each vertex. In the *realistic traffic light* problem, the number of traffic lights on each vertex is equal to its degree, and the drivers coming to the vertex from different road segments see different traffic lights. For the simplified version, we use $t(v)$ to denote the starting time of the green light at vertex v, while for the realistic version, we use $t(u, v)$ to denote the starting time of the green light at vertex v which is seen by drivers coming from u.

An example of a traffic network for the simplified traffic light problem is shown in Figure 1. There is a traffic light on every vertex (intersection). In the figure, the number on each vertex represents the starting time of the green light at that vertex.

We define a penalty for each edge (v_1, v_2) in E to reflect the goodness of the relative timing settings between $t(v_1)$ and $t(v_2)$ ($t(v_1, v_2)$ and $t(v_2, v_1)$ for the realistic version). We first use an example, as shown in Figure 2(a), to illustrate how to calculate the penalty. In the figure, the road segment (v_1, v_2) has a length of 6 units, which means that it takes 6 units of time for a car to pass through this road segment. Assume that the traffic light cycle is 20 units, the red light duration is

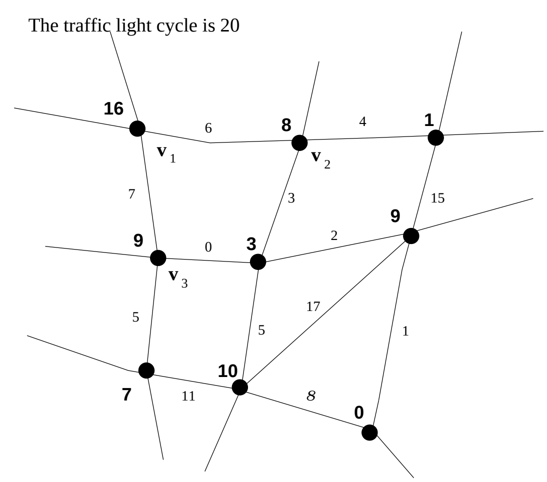

Figure 1: An example of a traffic network .

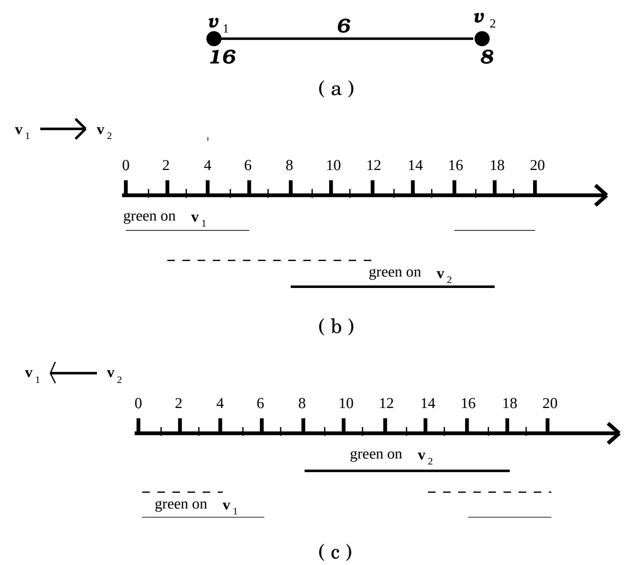

Figure 2: The penalty of an edge with the traffic light cycle 20. (a) The timing setting of an edge. (b)The penalty from v_1 to v_2. (c)The penalty from v_2 to v_1.

10 units, and the green light duration is 10 units. Suppose we set $t(v_1) = 16$ and $t(v_2) = 8$. Figure 2(b) shows the scenario of a car coming from v_1 to v_2. The thin lines represent the time intervals that the green light is on at vertex v_1, and hence, a car can pass through v_1 during these intervals. For a car passing v_1 during the green light interval $[-4, 6)$, the dotted line represents the time interval when it arrives at v_2. The thick line represents the time interval that the green light is on at v_2. It is easy to see that if a car passes through v_1 during the time interval $[2, 6)$, it will arrive at v_2 at the time interval $[8, 12)$. And since the green light is on at v_2 during this time interval, the car can pass v_2 without waiting. In contrast, if the car passes through v_1 during the time interval $[-4, 2)$, when it arrives at v_2 during the time interval $[2, 8)$, the red light is on at v_2, and hence, it must wait at v_2 before it can pass v_2. Therefore, we define the *penalty* from v_1 to v_2 to be the length of the waiting interval $[2, 8)$, which is $8 - 2 = 6$. Similarly, as shown in Figure 2(c), the penalty is defined to be 2 from v_2 to v_1. The total penalty for the road segment connecting v_1 and v_2 is thus 8. (6 for v_1 to v_2 and 2 for v_2 to v_1).

Let $c_{uv} =\mid t(v) - d(u,v) - t(u) \mid mod\ T$ if $(u,v) \in E$ and $c_{uv} = 0$ if $(u,v) \notin E$. It can be easily checked that in general, the penalty from u to v (with respect to the timing settings t) is $P_{uv} = \min\{c_{uv}, T - c_{uv}\}$. Note that P_{uv} may not be equal to P_{vu}. Since each edge has two directions, by definition, the penalty of edge (u,v) is equal to $P_{uv} + P_{vu}$.

Now, we give a formal definition for the *simplified traffic light* problem. Given a traffic light cycle T (T is even) and a graph $G = (V, E)$ in which each edge (u,v) is associated with a nonnegative integral distance $d(u,v)$, we want to assign $t(v)$ for each vertex $v \in V$ such that the total penalty $(\sum_{u \in V} \sum_{v \in V} P_{uv})$ is minimized.

For the realistic traffic light problem, the number of traffic lights on each vertex is equal to its degree, and each traffic light on a vertex corresponds to an edge incident to the vertex. We will also use (u,v) to denote the traffic light on vertex v to which the edge (u,v) is incident. We require that the difference between the starting times of the green lights of any two traffic lights on a vertex must be either 0 or $T/2$, and these differences are specified in the problem input. For each traffic light (u,v), we specify a value $F(u,v)$ in the input, where $F(u,v) = 0$ or 1. The input F values require that $\mid t(u,v) - t(w,v) \mid mod\ T = \mid F(u,v) - F(w,v) \mid \cdot T/2$, for all $(u,v), (w,v) \in E$.

The penalties of the edges in the realistic traffic light problem can be defined as follows. We assume that $|t(u,v) - t(w,v)| = 0$ or $T/2$. We assume there is no restriction between $t(u,v)$ and $t(v,u)$. Let $C_{uv} =\mid t(u,v) - d(u,v) - t(v,u) \mid mod\ T$ if $(u,v) \in E$, and $C_{uv} = 0$ if $(u,v) \notin E$. We define the penalty from u to v (with respect to the timing settings t) as $P_{uv} = \min\{C_{uv}, T - C_{uv}\}$. The goal of the realistic traffic light problem is then to find assignments $t(u,v)$ and $t(v,u)$ for each edge $(u,v) \in E$ such that the total penalty $(\sum_{u \in V} \sum_{v \in V} P_{uv})$ is minimized.

3 NP-Hardness of the Traffic Light Problem

To prove the NP-hardness of the traffic light problem, we first discuss some related problems. The conventional *vertex coloring* problem [1, 9] is defined as follows. Given a graph $G = (V, E)$, a *K-(vertex) coloring* of G is a mapping C from the vertex set V to a set of K colors. A graph G is *K-colorable* if there exists a coloring C with K colors for G such that no two adjacent vertices have the same color, i.e., if $(u,v) \in E$ then $C(u) \neq C(v)$. The vertex coloring problem is to ask, given a graph G and a positive integer K, if G is *K-colorable*. The vertex coloring problem has been proved to be NP-complete for all fixed $K \geq 3$ [4].

We generalize the vertex coloring problem to the following problem.

Problem 1 *Given a graph $G = (V, E)$ and a weight function $w_1 : E \to \Re$, find a K-coloring C that minimizes the following objective function:*

$$W_1(C) = \sum_{(u,v) \in E} \delta_1(u,v) \cdot w_1(u,v),$$

where

$$\delta_1(u,v) = \begin{cases} 1 & \text{if } C(u) = C(v) \\ 0 & \text{otherwise.} \end{cases} \qquad \square$$

Note that we allow the weight to be negative. If $w_1(u,v) = 1$ for all $(u,v) \in E$ and we ask if there exists a K-coloring C such that $W_1(C) = 0$, then the problem is equivalent to the conventional vertex coloring problem. This means that the vertex coloring problem is a special case of the unweighted version of Problem 1. Therefore, Problem 1 is also NP-hard for all fixed $K \geq 3$.

Now, consider the following problem.

Problem 2 *Given a graph $G = (V, E)$, let $w_2 : E \to \Re^+ \cup \{0\}$ be a weight function and $l : E \to \{0, 1\}$ a label function on E. The problem is to find a K-coloring C which minimizes the following objective function:*

$$W_2(C) = \sum_{(u,v) \in E} \delta_2(u,v) \cdot w_2(u,v),$$

where

$$\delta_2(u,v) = \begin{cases} 1 & \textit{if } l(u,v)=1 \textit{ and } C(u)=C(v), \\ & \textit{or if } l(u,v)=0 \textit{ and } C(u)\neq C(v) \quad \Box \\ 0 & \textit{otherwise.} \end{cases}$$

Intuitively, $l(u,v)=1$ means that u and v should be assigned different colors, otherwise (i.e., $C(u)=C(v)$) we impose a penalty $w_2(u,v)$ on edge (u,v). Similarly, $l(u,v)=0$ means that u and v should be assigned the same color, otherwise we impose a penalty $w_2(u,v)$ on the edge.

The following theorem proves that Problem 1 and Problem 2 are, in fact, equivalent.

Theorem 1 *Problem 1 and Problem 2 are equivalent.*

Another graph theory problem related to the traffic light problem is the *maximum cut problem* $[1,8,9]$. A *cut* of a graph $G=(V,E)$ is an edge set $\{(u,v) \mid (u,v)\in E, u\in V_1 \text{ and } v\in V_2\}$, where V_1 and V_2 is a partition of V, i.e., $V=V_1\cup V_2$ and $V_1\cap V_2=\emptyset$. The maximum cut problem is to find a cut that has the maximum cardinality. If each edge is also associated with a weight, and we want to find a cut that has the maximum total weight, the problem is called the *weighted maximum cut problem*. If we think that all vertices in V_1 are colored with one color, and all vertices in V_2 are colored by another color, then minimizing the total weight of assigning the same color to two adjacent vertices is equivalent to maximizing that of the cut. Thus, the weighted maximum cut problem is equivalent to Problem 1 with $K=2$.

Now, consider the simplified traffic light problem with $T=2$. Given a graph $G=(V,E)$ in which each edge (u,v) is associated with a nonnegative integral distance $d(u,v)$, if $d(u,v)$ is odd and $t(u)=t(v)$ then the penalty of (u,v) is $P_{uv}+P_{vu}=\min\{c_{uv},T-c_{uv}\}+\min\{c_{vu},T-c_{vu}\}=2$, where $c_{uv}=\mid t(v)-d(u,v)-t(u)\mid$ *mod* T. If $d(u,v)$ is odd and $t(v)\neq t(u)$ then the penalty of (u,v) is 0. On the other hand, if $d(u,v)$ is even and $t(v)=t(u)$ then the penalty of (u,v) is 0, and if $d(u,v)$ is even and $t(v)\neq t(u)$ then the penalty of (u,v) is 2. Therefore, if we let $l(u,v)=d(u,v)$ *mod* T and $w_2(u,v)=2$, it is easy to see that coloring the vertices in V by two colors with nonnegative weight (penalty) is equivalent to assigning $t(v)$ to the vertices. Thus, the simplified traffic light problem with $T=2$ is equivalent to the unweighted version of Problem 2 with $K=2$ which, in turn, is equivalent to the unweighted maximum cut problem. Shih *et al.* [8] gave an algorithm of time complexity $O(n^{\frac{3}{2}}\log n)$ to solve the maximum cut problem on a planar graph, where n is the number of vertices in the graph. Therefore, the simplified traffic light problem with $T=2$ on a planar graph can also be solved in $O((n^{\frac{3}{2}}\log n)$ time.

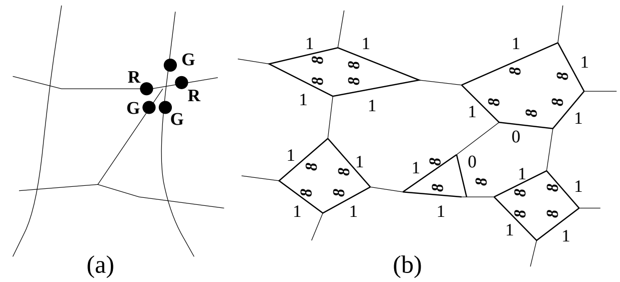

Figure 3: An example of the transformation from the realistic traffic light problem to Problem 2.

Moreover, the unweighted version of Problem 2 is a special case of the simplified traffic light problem with $T\geq 3$. Therefore, the simplified traffic light problem with $T\geq 3$ is NP-hard since the unweighted version of Problem 2 is NP-hard.

The realistic traffic light problem on planar graphs with $T=2$ can also be transformed to Problem 2. Since the number of traffic lights on a vertex is equal to its degree, we will transform each vertex v to a $d(v)$-polygon, where $d(v)$ is the degree of v. Given a planar graph $G=(V,E)$ with its plane realization in which each edge (u,v) is associated with a nonnegative integral distance $d(u,v)$, we construct a new planar graph $G'=(V',E')$, where $V'=\{v_{ij},v_{ji} \mid (v_i,v_j)\in E\}$, $E'=E_1'\cup E_2'$, $E_1'=\{(v_{ij},v_{ji}) \mid v_{ij},v_{ji}\in V'\}$, and $E_2'=\{(v_{ji},v_{ki}) \mid v_i,v_j,v_k\in V,(v_j,v_i) \text{ and } (v_k,v_i)$ are two neighboring edges in the plane realization of $G\}$. If $(v_{ij},v_{ji})\in E_1'$, then set $l(v_{ij},v_{ji})=d(v_i,v_j)\ mod\ T$ and $w_2(v_{ij},v_{ji})=1$. If $(v_{ji},v_{ki})\in E_2'$ and we require that $|t(v_{ji})-t(v_{ki})|\ mod\ T=0$, then set $l(v_{ji},v_{ki})=0$ and $w_2(v_{ji},v_{ki})=\infty$. If $(v_{ji},v_{ki})\in E_2'$ and we require that $|t(v_{ji})-t(v_{ki})|\ mod\ T=T/2$, then set $l(v_{ji},v_{ki})=1$ and $w_2(v_{ji},v_{ki})=\infty$. It can be shown that a 2-coloring with nonnegative weight (penalty) for the transformed graph corresponds to a valid timing settings for the traffic light problem, and a minimum-weight 2-coloring corresponds to a timing setting for the traffic light problem that has the minimum total penalty. An example for this transformation is shown in Figure 3. Again, using the maximum cut algorithm described in [8], the realistic traffic light problem on planar graphs with $T=2$ can be solved in time $O(n^{\frac{3}{2}}\log n)$.

4 Properties of the Traffic Light Problem

In this section, we discuss some properties of the simplified traffic light problem. Given a traffic network

$G = (V, E)$ and a traffic light cycle T (T is even), we assume, without loss of generality, that $d(u, v)$ and $t(v)$ are integers, $0 \leq d(u, v) \leq T - 1$, and $0 \leq t(v) \leq T - 1$, for all $v \in V$ and all $(u, v) \in E$ (note that all traffic lights have the same cycle T).

Recall that the penalty (with respect to t) from u to v is defined as $P_{uv} = \min\{c_{uv}, T - c_{uv}\}$, where $c_{uv} = \mid t(v) - d(u, v) - t(u) \mid mod\, T$. For convenience of discussion, we define the *penalty function* f_T as follows.

- $f_T(x + T) = f_T(x)$;
- $f_T(x) = x$, if $0 \leq x \leq T/2$;
- $f_T(x) = T - x$, if $T/2 < x \leq T$.

Using the penalty function f_T, the penalty P_{uv} from u to v can now be denoted as $f_T(t(v) - d(u, v) - t(u))$. For example, in Figure 2(a), the penalty from v_1 to v_2 is $P_{12} = f_{20}(8 - 6 - 16) = f_{20}(-14) = f_{20}(6) = 6$ and the penalty from v_2 to v_1 is $P_{21} = f_{20}(16 - 6 - 8) = 2$. Thus, the penalty of edge (v_1, v_2) is equal to $P_{12} + P_{21} = 8$.

We define the *extra penalty* of an edge as the actual penalty under a certain setting minus the minimum penalty of the edge. For example, in Figure 2(a), if we set $t(v_1) = 16$ and $t(v_2) = 11$, then the penalty of (v_1, v_2) is equal to $P_{12} + P_{21} = f_{20}(11 - 6 - 16) + f_{20}(16 - 6 - 11) = 9 + 1 = 10$. However, the minimum penalty of (v_1, v_2) is equal to 8 (A setting method will be given later.). Thus, the extra penalty of this edge under this setting is $10 - 8 = 2$.

We now prove some properties of the penalty function f_T in the following theorem.

Theorem 2 **(1)** $f_T(f_T(x)) = f_T(x)$.

(2) $f_T(-x) = f_T(x)$.

(3) $f_T(x + T/2) = f_T(x - T/2) = T/2 - f_T(x)$.

(4) $f_T(x + y) \leq f_T(x) + f_T(y)$.

Next, we consider a special kind of timing setting method which requires that the penalties of both directions of every edge are the same, that is, $P_{uv} = P_{vu}$ for each $(u, v) \in E$. For $(v_1, v_2) \in E$, given a traffic light cycle T, if we set $t(v_1) = t(v_2) = k$, $0 \leq k < T$, then $P_{12} = f_T(t(v_2) - d(v_1, v_2) - t(v_1)) = f_T(k - d(v_1, v_2) - k) = f_T(d(v_1, v_2)) = P_{21}$. And if we set $t(v_1) = k$ and $t(v_2) = (k + T/2)\ mod\ T$, $0 \leq k < T$, then $P_{12} = f_T(k + T/2 - d(v_1, v_2) - k) = f_T(d(v_1, v_2) - T/2)$ (by Theorem 2), and $P_{21} = f_T(k - d(v_1, v_2) - k - T/2) = f_T(d(v_1, v_2) + T/2)$, thus $P_{12} = P_{21}$ (by Theorem 2). Hence, if we set $t(v_1) = k$, to get the same penalties for both directions of (v_1, v_2), it can be easily shown that we have only two choices to set $t(v_2)$: k or $(k + T/2)\ mod\ T$. We call this simplified traffic light problem with the

Table 1: The smaller penalty on an edge (v_1, v_2).

condition	$t(v_1)$	$t(v_2)$	smaller penalty
$0 \leq d(v_1, v_2) \leq \frac{T}{4}$	k	k	$2d(v_1, v_2)$
$\frac{T}{4} < d(v_1, v_2) \leq \frac{T}{2}$	k	$k + \frac{T}{2}$	$T - 2d(v_1, v_2)$
$\frac{T}{2} < d(v_1, v_2) \leq \frac{3T}{4}$	k	$k + \frac{T}{2}$	$2d(v_1, v_2) - T$
$\frac{3T}{4} < d(v_1, v_2) \leq T$	k	k	$2T - 2d(v_1, v_2)$

Table 2: The larger penalty on an edge (v_1, v_2).

condition	$t(v_1)$	$t(v_2)$	larger penalty
$0 \leq d(v_1, v_2) \leq \frac{T}{4}$	k	$k + \frac{T}{2}$	$T - 2d(v_1, v_2)$
$\frac{T}{4} < d(v_1, v_2) \leq \frac{T}{2}$	k	k	$2d(v_1, v_2)$
$\frac{T}{2} < d(v_1, v_2) \leq \frac{3T}{4}$	k	k	$2T - 2d(v_1, v_2)$
$\frac{3T}{4} < d(v_1, v_2) \leq T$	k	$k + \frac{T}{2}$	$2d(v_1, v_2) - T$

above timing setting restriction the *two-phase setting problem*, since there are only two possible "phases," $k\ mod\ T$ and $|k + T/2|\ mod\ T$, for the timing settings of all vertices in V. Note that the total penalty is independent of the actual value of k.

Tables 1 and 2 list the assignments for $t(v_2)$ that will give the smaller and the larger values of $P_{12} + P_{21} = 2P_{12}$, respectively.

It is easy to check that the following transformation from the two-phase setting problem to Problem 2 with $K = 2$ is valid. Let $l(v_1, v_2) = 1$ if $T/4 < d(v_1, v_2) \leq 3T/4$ and $l(v_1, v_2) = 0$ otherwise. Let $w_2(v_1, v_2) = 2 \cdot |f_T(t(k) - d(v_1, v_2) - t(k)) - f_T(t(k) - d(v_1, v_2) - t(k + T/2))|$. Therefore, again, we can solve the two-phase setting problem on planar graphs in $O(n^{\frac{3}{2}} \log n)$ time by the algorithm designed for the weighted maximum cut problem.

Note, however, that we may not always get the minimum total penalty using the two-phase setting method. The following lemma gives a method to find the minimum penalty of an edge.

Lemma 1 *Given an edge (v_1, v_2) with distance $d(v_1, v_2)$, the minimum penalty of (v_1, v_2) is equal to $f_T(2d(v_1, v_2))$ and the maximum penalty of (v_1, v_2) is equal to $T - f_T(2d(v_1, v_2))$. Besides, the minimum penalty can be obtained if the penalty of one direction of the edge is set to zero.*

Lemma 2 *Given a graph $G = (V, E)$, where $V = \{v_i \mid 1 \leq i \leq n + 1\}$ and $E = \{\ (v_i, v_{i+1})\ |1 \leq i \leq n\}$ (G is a path of $n + 1$ vertices), the minimum penalty of G can be obtained by setting either $P_{ij} = 0$ or $P_{ji} = 0$ for every edge and the minimum penalty of G is equal to*

$\sum_{i=1}^{n} f_T(2d(v_i,v_{i+1}))$ *and the maximum penalty of* G *is equal to* $n \cdot T - \sum_{i=2}^{n} f_T(2d(v_i,v_{i+1}))$.

In fact, the minimum penalty of an edge (u,v) does not occur only at $P_{uv}=0$ or $P_{vu}=0$. Lemma 3 shows the solution is a range dependent on $d(u,v)$.

Lemma 3 *Given an edge* (v_1,v_2) *with distance* $d(v_1,v_2)$, *if* $t(v_1)$ *has been set, then we will get the minimum penalty of* (v_1,v_2) *when* $t(v_2)$ *is any integer number between* $f_T(t(v_1)-d(v_1,v_2))$ *and* $f_T(t(v_1)+d(v_1,v_2))$. *In other words, the number of solutions of* $t(v_2)$ *for minimizing the penalty of* (v_1,v_2) *is* $f_T(2d(v_1,v_2))+1$.

Theorem 3 will give a method to find the minimum penalty of two adjacent edges when the two endpoints have been set.

Theorem 3 *Given two adjacent edges* (v_1,v_2) *and* (v_2,v_3), *suppose* $t(v_1)$ *and* $t(v_3)$ *have been set, and* $f_T(2d(v_1,v_2)) \leq f_T(2d(v_2,v_3))$. *The minimum penalty for these two edges can be found if we try to assign each integer between* $f_T(t(v_1)-d(v_1,v_2))$ *and* $f_T(t(v_1)+d(v_1,v_2))$ *to* $t(v_2)$. *Besides, the setting of* $t(v_2)$ *for either* $P_{12}=0$, *i.e.* $t(v_2)=f_T(t(v_1)+d(v_1,v_2))$, *or* $P_{21}=0$, *i.e.* $t(v_2)=f_T(t(v_1)-d(v_1,v_2))$, *is in the range. The maximum penalty for these two edges can be found if we try to assign each integer between* $f_T(t(v_1)-d(v_1,v_2)+T/2)$ *and* $f_T(t(v_1)+d(v_1,v_2)+T/2)$ *to* $t(v_2)$. *Besides, the setting of* $t(v_2)$ *for either* $P_{12}=T/2$, *i.e.* $t(v_2)=f_T(t(v_1)+d(v_1,v_2)+T/2)$, *or* $P_{21}=T/2$ *i.e.,* $t(v_2)=f_T(t(v_1)-d(v_1,v_2)+T/2)$, *is in the range. And, the maximum penalty plus the minimum penalty is equal to* $2T$.

We extend Theorem 3 to a graph consisting of a path with n edges.

Theorem 4 *Given a graph* $G=(V,E)$, *where* $V=\{v_i \mid 1\leq i\leq n+1\}$, $E=\{(v_i,v_{i+1}) \mid 1\leq i\leq n\}$ *and* $d_i=d(v_i,v_{i+1})$, $1\leq i\leq n$, *under the condition that* $t(v_1)$ *and* $t(v_{n+1})$ *have been set, if* $f_T(2d_n)\geq max\{f_T(2d_1), f_T(2d_2), \cdots, f_T(2d_{n-1})\}$, *then the minimum penalty of* G *can be found by testing the solution set* S, *where* S *consists of all possible settings such that* (v_1,v_2), (v_2,v_3), $\cdots$, (v_{n-2},v_{n-1}) *and* (v_{n-1},v_n) *have zero extra penalty.*

In Theorem 4, if we combine the starting vertex and the ending vertex of a path into a single vertex, we will get a polygon. Thus, we have a similar property for a polygon as follows.

Corollary 1 *Given a graph* $G=(V,E)$, *where* $V=\{v_i \mid 1\leq i\leq n\}$, $E=\{(v_1,v_2), (v_2,v_3), \cdots, (v_{n-1},v_n), (v_n,v_1)\}$, $d_i=d(v_i,v_{i+1})$, $1\leq i\leq n-1$ *and* $d_n=d(v_n,v_1)$, *if* $f_T(2d_n)\geq max\{f_T(2d_1), f_T(2d_2), \cdots, f_T(2d_{n-1})\}$, *then the minimum penalty of* G *can be found by testing the solution set* S, *where* S *consists of all possible settings such that* (v_1,v_2), (v_2,v_3), $\cdots$, (v_{n-2},v_{n-1}) *and* (v_{n-1},v_n) *have zero extra penalty.*

For the realistic traffic light problem, the number of traffic lights on each vertex is equal to its degree. In this problem, we assume the difference of the starting times of the green lights of different traffic lights on the same vertex is either 0 or $\frac{T}{2}$.

Theorem 5 shows that the realistic traffic light problem on a polygon can be reduced to the simplified traffic light problem.

Theorem 5 *Given a graph* $G=(V,E)$, *where* $V=\{v_i|1\leq i\leq n\}$, $E=\{(v_1,v_2), (v_2,v_3), \cdots, (v_{n-1},v_n), (v_n,v_1)\}$, $d_i=d(v_i,v_{i+1})$, $1\leq i\leq n-1$ *and* $d_n=d(v_n,v_1)$, *the realistic traffic light problem can be reduced to the simplified traffic light problem.*

5 A Heuristic Algorithm

First, we will give a theorem to show that the minimum penalty of a polygon with n edges can be found in $O(n)$ time. Before giving the theorem, we have the following lemma.

Lemma 4 *Let the minimum penalty of a traffic network* $G=(V,E)$ *with traffic light cycle* T *be* P. *For a particular vertex* $v_1\in V$, *if* $t(v_1)=m$, *for any* m *and* $0\leq m\leq T-1$, *there exists a timing setting method whose total penalty is* P.

Now, we will propose a heuristic algorithm with polynomial time to solve the simplified traffic light problem.

Let $G=(V,E)$ be a graph and the distance of each edge $(u,v)\in E$ be $d(u,v)$. Note that the minimum penalty of each edge (u,v) is $f_T(2d(u,v))$. Our heuristic algorithm will find a near minimum penalty of G as follows.

Algorithm Algorithm Traffic Lights Setting(*TLS*)

Input: A traffic light cycle T, where T is an integer and $T\geq 3$, and $G=(V,E)$ where $V=\{v_i|1\leq i\leq n\}$ and every edge e is associated with an integer distance $d(e)$, $0\leq d(e)\leq T-1$.

Output: $t(v)$ for each $v \in V$ such that the penalty P of G for the simplified traffic light problem is minimized as possible as.

Step 1: Set $t(v_1) = 0$.

Step 2: Set $w(v_i, v_j) = f_T(2d(v_i, v_j))$ as the weight of each edge $(v_i, v_j) \in E$. In G, find a minimum spanning tree rooted at v_1.

Step 3: $a_i = (t(v_j) - d(v_i, v_j))$ mod T and $b_i = t(v_j) + d(v_i, v_j))$ mod T, where v_j is the father of v_i in the rooted minimum spanning tree.

Step 4: Set $t(v_i) = a_i$, $2 \leq i \leq n$. Let P denote the penalty of G under this setting.

Step 5: For each leaf node v_i do
For each integer x in the range between a_i and b_i.
Let P' be the penalty of G when $t(v_i) = x$.
If $P > P'$ then
let $P = P'$ and set $t(v_i) = x$.

end

Algorithm TLS guarantees that each edge on the minimum spanning tree has zero extra penalty. Why does this setting method get a good solution? There are two reasons. First, by Theorem 4, in a polygon(cycle), we can find the minimum penalty in the settings that make all edges, except the one with the largest weight $f_T(2d(e))$, have zero extra penalty. Second, by Lemma 1, for an edge, the sum of the minimum penalty and the maximum penalty is equal to T. In other words, an edge with smaller minimum penalty may have larger extra penalty. For a leaf node v_i on the rooted minimum spanning tree, each integer x in the range between a_i and b_i achieves the goal that each edge on the minimum spanning tree has zero extra penalty. Thus, the job of Step 5 is to do fine tuning to get a better solution. Clearly, the time required for the algorithm is $O(n^2)$, which is spent to find the minimum spanning tree, plus the time needed for Step 5.

The realistic traffic light problem can be easily transformed into the simplified traffic light problem, as done in Section 3.

6 Experiment Results

As noted earlier, it is hard to find the minimum penalty for a traffic network. Thus, we will compare our experiment results of our heuristic algorithm for the simplified traffic light problem with the lower bound of penalty and some other simpler heuristic algorithms.

Consider the traffic network which is a general graph $G = (V, E)$. For each edge (u, v) in G, $f_T(2d(u, v))$ is the minimum penalty of (u, v). We define the lower bound of the penalty of G as the sum of the minimum penalties of all edges in G.

Table 3 shows the experiment results of seven general graphs of different sizes. In each experiment, shown in one row, 50 cases are tested and the average penalty of them is calculated. The results shown in Table 3 are obtained by four different methods. The second column reports the lower bound by our definition, the third column shows the penalty obtained by algorithm TLS, the fourth column shows the result obtained by tree optimal method and the fifth column shows the penalty when all traffic lights are set randomly. In the tree optimal method, we first find a depth-first spanning tree and then set each edge in the spanning tree to have zero extra penalty. We have to point out that the real minimum penalty should be larger than the lower bound and smaller than our heuristic result.

Table 4 shows the experiment results of seven planar graphs of different sizes.

7 Conclusions

In a traffic network, the importance for different roads is different. Thus, another factor for setting traffic lights is the traffic flow on the roads. When the traffic flow is involved, the problem becomes the weighted traffic light problem. Now, we give a formal definition for the weighted traffic light problem. Given a traffic light cycle T, where T is even, and a graph $G = (V, E)$ in which each edge (u, v) is associated with a distance $d(u, v)$ and the traffic flow $w(u, v)$, we want to assign $t(v)$ for each vertex $v \in V$. Let $P_{uv} = min\{c_{uv}, T - c_{uv}\}$, where $c_{uv} = \mid t(v) - d(u, v) - t(u) \mid$ mod T if $(u, v) \in E$ and $c_{uv} = 0$ if $(u, v) \notin E$. Our goal is to assign $t(v)$ for each $v \in V$ such that the total penalty $(\sum_{u \in V} \sum_{v \in V} w(u, v) \cdot P_{uv})$ is minimized. This weighted traffic light problem can also be solved by our heuristic algorithm. For the weighted traffic light problem, if every traffic light can get only the traffic flow of neighboring vertices, an efficient distributed algorithm to find the minimum penalty of a traffic network may be preferred.

In our traffic light problem, we find the total minimum penalty of a traffic network. In fact, a driver cares how to arrive at the destination as soon as possible. Consider a graph $G = (V, E)$, where $V = \{v_1, v_2, v_3, v_4, v_5\}$ and $E = \{ (v_i, v_{i+1}) \mid 1 \leq i \leq 4\}$ (G is a path of 5 vertices) and $d(v_1, v_2) = 3$, $d(v_2, v_3) = 1$,

Table 3: The experiment results on a general graph consisting of n vertices.

n	lower bound	TLS	tree optimal	random
10	135.80	195.44 (143.92%)	226.28 (166.63%)	268.52 (197.73%)
25	361.96	527.12 (145.63%)	609.40 (168.36%)	734.84 (203.02%)
50	746.84	1079.20 (144.50%)	1261.44 (168.90%)	1496.36 (200.36%)
80	1184.30	1661.00 (140.25%)	1972.05 (166.52%)	2356.30 (198.96%)
100	1482.50	2083.95 (140.57%)	2461.55 (166.04%)	2976.55 (200.78%)
200	2977.75	4211.90 (141.45%)	4973.80 (167.03%)	5942.60 (199.57%)
500	7505.60	10669.20 (142.15%)	12470.60 (166.15%)	15083.60 (200.96%)

Table 4: The experiment results on a planar graph consisting of n vertices.

n	lower bound	TLS	tree optimal	random
10	77.44	91.72 (118.44%)	110.80 (143.08%)	151.32 (195.40%)
25	212.76	250.44 (117.71%)	306.08 (143.86%)	419.88 (197.35%)
50	420.00	504.36 (120.09%)	605.68 (144.21%)	857.68 (204.21%)
80	688.24	820.92 (119.28%)	984.56 (143.05%)	1379.56 (200.45%)
100	835.30	1003.70 (120.16%)	1211.40 (145.03%)	1727.80 (206.85%)
200	1751.00	2082.40 (118.93%)	2434.60 (139.04%)	3436.60 (196.26%)
500	4440.80	5243.60 (118.08%)	6258.40 (140.93%)	8716.00 (196.27%)

$d(v_3, v_4) = 4$, $d(v_4, v_5) = 5$. If the traffic light cycle is 20, then we can get the minimum penalty of G, which is 26, when $t(v_1) = 0$, $t(v_2) = 3$, $t(v_3) = 4$, $t(v_4) = 8$ and $t(v_5) = 13$. In fact, if a driver wants to drive from v_1 to v_5 , then the driver needs to wait only 10 unit of times at most, no matter what time the driver starts from v_1. And if a driver drives from v_5 to v_1 , then the driver needs to wait 10 unit of times at most, too. Thus, there may be another way to model the traffic light problem. And an algorithm is needed to be designed to get the shortest time from some special source points to some special destination points in a traffic network.

References

[1] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications.* The Macmillan Press LTD, 1976.

[2] G. Casadei, A. Palareti, and G. Proli, "Classifier system in traffic management," *Proc. of the International Conference on Artificial Neural Nets and Genetic Algorithms*, Innsbruck, Austria, pp. 620–627, Apr. 14-16 1993.

[3] J. Favilla, A. Machion, and F. Gomide, "Fuzzy traffic control: adaptive strategies," *Proc. of the Second IEEE International Conference on Fuzzy Systems*, San Francisco, USA, pp. 506–511, Mar. 28 - Apr. 10 1993.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability a Guide to the Theory of NP-Completeness.* San Francisco, USA: Freeman, 1979.

[5] P. Gassman, "Bustra-lsa-ars (level crossing control)," *Signal und Draht*, Vol. 70, pp. 131–134, Apr. 1993.

[6] I. McKendrick, "Applications of fuzzy logic control," *Proc. of ISA International England Section Conference, Advances in Control II, The Application of Advanced and Expert Systems for On-Line Process Control*, Birmingham, UK, p. 15, Apr. 29 1992.

[7] T. Riedel and U. Brunner, "A control algorithm for traffic lights," *Proc. of the First IEEE Conference on Control Applications*, Dayton, OH, USA, pp. 901–906, Sep. 13-16 1992.

[8] W. K. Shih, S. Wu, and Y. S. Kuo., "Unifying maximum cut and minimum cut of a planar graph," *IEEE Transactions on Computers*, Vol. 39, No. 5, pp. 694–697, May 1990.

[9] R. E. Tarjan, *Data Structures and Network Algorithms.* Pennsylvania: Society for Industrial and Applied Mathematics, 1983.