J Comb Optim
DOI 10.1007/s10878-013-9614-z

Generalized Canadian traveller problems

Chung-Shou Liao · Yamming Huang

Abstract This study investigates a generalization of the CANADIAN TRAVELLER PROBLEM (CTP), which finds real applications in dynamic navigation systems used to avoid traffic congestion. Given a road network $G = (V, E)$ in which there is a source s and a destination t in V, every edge e in E is associated with two possible distances: original $d(e)$ and jam $d^+(e)$. A traveller only finds out which one of the two distances of an edge upon reaching an end vertex incident to the edge. The objective is to derive an adaptive strategy for travelling from s to t so that the competitive ratio, which compares the distance traversed with that of the static s, t-shortest path in hindsight, is minimized. This problem was initiated by Papadimitriou and Yannakakis. They proved that it is PSPACE-complete to obtain an algorithm with a bounded competitive ratio. In this paper, we propose tight lower bounds of the problem when the number of "traffic jams" is a given constant k; and we introduce a deterministic algorithm with a $\min\{r, 2k+1\}$-ratio, which meets the proposed lower bound, where r is the worst-case performance ratio. We also consider the Recoverable CTP, where each blocked edge is associated with a recovery time to reopen. Finally, we discuss the uniform jam cost model, i.e., for every edge e, $d^+(e) = d(e) + c$, for a constant c.

Keywords Canadian traveller problem · Competitive ratio · Online algorithm

---

An extended abstract of this paper appeared in the 23rd International Symposium on Algorithms and Computation (ISAAC 2012).

C.-S. Liao (✉) · Y. Huang
Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan
e-mail: csliao@ie.nthu.edu.tw

Y. Huang
e-mail: s100034528@m100.nthu.edu.tw

Published online: 23 April 2013

1 Introduction

Consider a road map $G = (V, E)$ represented by a set V of vertices connected by edges, where each edge $e \in E$ is associated with the time it takes for the traveller to traverse the edge. From an online perspective, the traveller is aware of the entire structure of the road network in advance; however, while some edges may be blocked by accidents during the trip, the problem will only become evident when the traveller reaches an end vertex incident to the blocked edge. This problem, called the CANADIAN TRAVELLER PROBLEM (CTP), and its variants can be considered as a natural assumption of some online routing problems. The CTP was defined by Papadimitriou and Yannakakis in 1991. Its objective is to design an efficient routing policy from a source to a destination under this condition of uncertainty. The major difficulty in developing a good strategy based on partial information is decision making about future blockages without any predictable traffic conditions. The CTP is actually a two-player game between a traveller and a malicious adversary who sets up road blockages in order to maximize the gap between the performance of the online strategy and that of the offline optimum (with the blocked edges removed). The criterion for measuring the quality of an online strategy is usually the competitive ratio of the algorithm Borodin and El-Yaniv (1998); Sleator and Tarjan (1985). The competitive ratio can be defined as follows: for any instance, the total cost of the online algorithm is at most its ratio times that of the optimal offline approach (under complete information). We will provide the formal definition later in this paper. Papadimitriou and Yannakakis (1991) showed that it is PSPACE-complete for the CTP to devise a strategy that guarantees a bounded competitive ratio.

For several years, there has been no significant progress in the development of competitive algorithms for this problem. Bar-Noy and Schieber (1991) explored several variations of the CTP from the worst-case scenario perspective, where the objective is to find a static (offline) algorithm that minimizes the maximum travel cost Ben-David and Borodin (1994). They considered the k-CTP, in which the number of blockages is bounded from below by a given constant k. Note that for an arbitrary k, the problem of designing a strategy that guarantees a given travel time remains PSPACE-complete, as shown in Bar-Noy and Schieber (1991); Papadimitriou and Yannakakis (1991). In addition, Bar-Noy and Schieber discussed the Recoverable k-CTP, where each blocked edge is associated with a recovery time, which is not very long to reopen relative to the traversed time. They also studied the stochastic model, where an independent blockage probability for each edge is given in advance. This model, which tries to minimize the expected ratio to the offline optimum, is known to be $\sharp$P-hard Papadimitriou and Yannakakis (1991). Subsequently, Karger and Nikolova (2008) investigated the stochastic CTP in special graph classes, and developed exact algorithms by applying techniques from the theory of Markov Decision Processes.

Recently, Westphal (2008) proved that there are no deterministic online algorithms within a $(2k + 1)$-competitive ratio for the k-CTP. The author designed a simple *reposition* algorithm that satisfies the lower bound, and also proposed a lower bound of $k + 1$ for the competitive ratio of any randomized online algorithms. Xu et al. (2009) developed two deterministic adaptive policies: a *greedy* strategy as well as a *comparison* strategy that incorporates the concept of reposition. The latter strategy also achieves the tight lower bound.

In this paper, we study a natural generalization of CTP, called the DOUBLE- VALUED GRAPH problem, which was initiated by Papadimitriou and Yannakakis (1991). Given a graph $G = (V, E)$ with a source s and a destination t in V, each edge e in E is associated with two possible distances: original $d(e)$ and jam $d^+(e)$, where d, $d^+ : E \to R^+$ and $d(e) < d^+(e)$, for each $e \in E$. A traveller only learns about the distance cost ($d(e)$ or $d^+(e)$) of an edge e on arrival at one of its end vertices. The goal is to develop an adaptive strategy for traversing the graph from s to t with incomplete information about traffic conditions, so that the competitive ratio is minimized. This problem is also PSPACE-complete, as shown by a reduction from quantified SAT (QSAT) Papadimitriou and Yannakakis (1991). Here, for a graph G without any jammed edges, i.e., $d^+(e) = d(e)$, for any $e \in E$, the distance of the s, t-shortest path is denoted by $d(s, t)$. On the other hand, let the s, t-shortest path P consist of ℓ edges, and consider the distance of the path P in the worst case during the trip, denoted by $d^+(s, t, k)$. More precisely, given a constant bound k of the number of jammed edges, if $\ell < k$, $d^+(s, t, k) = d(s, t) + \sum_{e \in P}(d^+(e) - d(e))$; otherwise, let E' consist of the k jammed edges that have the maximum jam costs in P. Then, $d^+(s, t, k) = d(s, t) + \sum_{e \in E'}(d^+(e) - d(e))$.

Our contribution. The main results of this study are detailed below.

1. We provide tight lower bounds for deterministic and randomized algorithms for DOUBLE- VALUED GRAPH in terms of k and $r = \frac{d^+(s,t,k)}{d(s,t)}$ when the number of traffic jams is up to a given constant k.
2. We present a deterministic adaptive strategy with a $\min\{r, 2k + 1\}$-competitive ratio that meets the proposed lower bound. The algorithm can also be applied directly to the Recoverable k-CTP that assumes the blocked edges are not found to be blocked again.
3. Finally, we study the uniform jam cost model of this problem, i.e., for every edge e, $d^+(e) = d(e) + c$, for a constant c, and derive a tight lower bound with an additive competitive ratio.

2 Preliminaries

We consider the DOUBLE- VALUED GRAPH problem with at most k traffic jams. Given a connected graph $G = (V, E)$ with a source s and a destination t, we denote the sequence of traffic jams in E learned by an online algorithm A during the trip as $S_i^A = (e_1, e_2, \ldots, e_i)$, where $1 \le i \le k$. Let $E_i^A = \{e_1, e_2, \ldots, e_i\} \subseteq E$, $1 \le i \le k$, consist of these jammed edges, and let E_k be the set of all jammed edges. In the following, the superscript A may be omitted without causing confusion. In addition, let $d : E \to R^+$ be the original distance function. The (traffic) jam distance function is $d^+ : E \to R^+$; that is, for each edge $e = (u, v) \in E$, $d(u, v) < d^+(u, v)$. Moreover, in the online problem, let $d_{E_i^A}(s, t)$ denote the travel cost from s to t, derived by an adaptive algorithm A that learns about traffic jam information E_i during the trip; and let $d_{E_k}(s, t)$ be the offline optimum from s to t under complete information E_k. We immediately have the following property Xu et al. (2009), where

$E_1 \subseteq E_2 \subseteq \ldots \subseteq E_k$.

$$d(s,t) \leq d_{E_1}(s,t) \leq \cdots \leq d_{E_k}(s,t). \tag{1}$$

We refer to Borodin and El-Yaniv (1998); Sleator and Tarjan (1985) and formally define the competitive ratio as follows: an online algorithm A is c^A-*competitive* for the DOUBLE- VALUED GRAPH problem if for any instances,

$$d_{E_i^A}(s,t) \leq c^A \cdot d_{E_k}(s,t) + \varepsilon, \quad 1 \leq i \leq k,$$

where c^A and ε are constants.

Similar to the proof in Westphal (2008), we propose tight lower bounds for DOUBLE- VALUED GRAPHS when we use deterministic and randomized algorithms, respectively.

Lemma 1 *For the* DOUBLE- VALUED GRAPH *problem, there is no deterministic online algorithm with a competitive ratio less than* $\min\{r, 2k+1\}$ *when the number of jammed edges is up to a given constant k.*

Proof Consider the graph in Fig. 1. Each edge is associated with two possible values. The first indicates the edge's original distance, and the second represents its jammed distance. The traveller has two ways to walk from s to t: (1) The traveller traverses a jammed edge. (2) The traveller is prevented from passing through jammed edges to reach t; that is, when the traveller learns about a jammed edge, he/she always selects another way if there is still a route without jams. Because there are $k+1$ different s, t-paths, the second strategy can work in this example. If the traveller traverses a jammed edge, then the distance cost is at least $1+\varepsilon^+$. Otherwise, the traveller must choose a path without jammed edges to traverse. This policy results in a distance cost of $k \cdot (1+1) + 1 + \varepsilon$ in the worst case; that is, the traveller would return to s exactly k times. Because the distance cost of the optimal path is $1+\varepsilon$, the competitive ratio of an arbitrary deterministic algorithm is not less than $\min\{\frac{1+\varepsilon^+}{1+\varepsilon}, \frac{2k+1+\varepsilon}{1+\varepsilon}\}$. When ε is sufficiently small, the ratio is at most $\min\{r, 2k+1\}$, where $r = \frac{d^+(s,t,k)}{d(s,t)} = \frac{1+\varepsilon^+}{1+\varepsilon}$. □

Next, given the independent probabilities of traffic congestion on all the edges, we consider randomized strategies to solve this online problem.

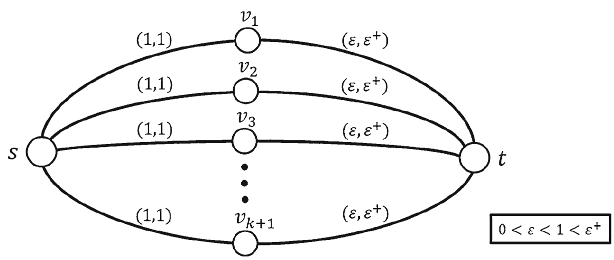

Fig. 1 An example of the lower bounds in the DOUBLE- VALUED GRAPH problem

Lemma 2 *For the* Double- valued Graph *problem, there is no randomized online algorithm with a competitive ratio less than* $\min\{r, k+1\}$ *when the number of jammed edges is up to a given constant k.*

Proof The example in Fig. 1 also illustrates this lower bound. Similarly, the traveller has two ways to traverse a path from s to t. If the traveller walks through a jammed edge, the expected distance cost is $1+\varepsilon^+$ because each of the $k+1$ s,t-paths has the same setting. Otherwise, the traveller must select another way to avoid jams when he/she finds a jammed edge (v_j, t) at v_j. The traveller would go back to s and then randomly choose (s, v_i), $v_i \in \{v_1, \ldots, v_{k+1}\} \setminus \{v_j\}$. Without loss of generality, we assume the traveller visits the vertices from v_1 to v_{k+1}, i.e., $S_j = ((v_1, t), (v_2, t), \ldots, (v_j, t))$, $1 \le j \le k+1$. We denote the probability that the traveller learns about the jammed edge (v_j, t) at v_j as $p(v_j)$. It is not hard to determine that $p(v_i|S_{i-1}) \ge \frac{k-i+1}{k-i+2}$, $1 < i \le k$, where $p(v_1) \ge \frac{k}{k+1}$ and $p(v_{k+1}|S_k) = 0$, so the expected distance cost can be computed as follows:

$$2 \times \sum_{i=1}^{k}\left(\prod_{j=1}^{i} p(v_j|S_{j-1})\right) + (1+\varepsilon) \ge 2\left(\frac{k}{k+1} + \frac{k-1}{k+1} + \ldots + \frac{1}{k+1}\right) + (1+\varepsilon)$$
$$= k+1+\varepsilon.$$

The distance cost of the optimal path is $1+\varepsilon$; thus, when ε is sufficiently small, the competitive ratio is not less than $\min\{\frac{1+\varepsilon^+}{1+\varepsilon}, \frac{k+1+\varepsilon}{1+\varepsilon}\} = \min\{r, k+1\}$. □

Note that the example in Fig. 1 also achieves the tightness of the two lower bounds. Based on the proofs of Lemmas 1 and 2, we introduce two strategies: the *greedy algorithm* and the *reposition algorithm* Westphal (2008); Xu et al. (2009), which will be used later. We denote them as GA and RA, respectively.

Greedy Algorithm (GA): Starting at a vertex v (including the source s), the traveller selects the shortest path from v to t by using Dijkstra's algorithm Dijkstra (1959) in a greedy manner, based on the current information E_i; that is, the distance cost of the path is $d_{E_i}(v, t)$. Note that if all the k jammed edges are known at the outset, the cost of the path derived by GA from the source s is the same as that of the offline optimum, $d_{E_k}(s, t)$.

Reposition Algorithm (RA): The traveller begins at the source s and follows the s,t-path with the cost $d(s, t)$. When the traveller learns about a jammed edge on the path to t, he/she returns to s and takes the s,t-path with the cost $d_{E_i}(s, t)$ based on the current information E_i. The traveller repeats this strategy until he/she arrives at t.

3 Double-valued graph

In this section, we present a deterministic algorithm called GR, which combines GA with RA to solve the Double- valued Graph problem. Its ratio meets the proposed lower bound. For convenience, let $e_i = (v_i, v_{i'})$, $1 \le i \le k$ be a jammed edge learned of by the traveller; and let v_i be the first end vertex of the jammed edge e_i that the traveller visits during the trip.

Algorithm 1: Greedy & Reposition Algorithm (GR)

Input : $G = (V, E), d : E \to R^+, d^+ : E \to R^+$ and a constant k;
Output : A route from s to t;

```
Initialize i = 0 and let v_0 = s and E_0 = ∅;
while the traveller does not arrive at t do
    Let r = d^+(v_i, t, k−i) / d_{E_i}(s, t);
    if r ≤ 2(k − i) + 1 then
        the traveller follows a path from v_i to t derived by GA;      ▷ the first strategy
    else
        if d_{E^{GR}_{i−1}}(s, t) + d_{E_i}(v_i, t) ≤ (i + 1) · d_{E_i}(s, t) then
            the traveller moves from v_i to t via GA until he/she finds a jammed edge;      ▷ the second strategy
        else
            the traveller moves from v_i to t via RA until he/she finds a jammed edge;      ▷ the third strategy
        end if
    end if
    Let i = i + 1 and let the new jammed edge be e_i = (v_i, v_{i'}) during the trip;
end while
```

This algorithm mainly consists of three routing policies when the traveller arrives at v_i. The first strategy is to follow a path from v_i to t derived by GA, irrespective of whether the traveller finds a new jammed edge. The second strategy for the traveller is to move from v_i to t via GA until he/she finds a jammed edge; and the last strategy is to move from v_i to t via RA until the traveller finds a jammed edge. Note that the ratio r has to be updated while the traveller is learning about a new jammed edge. In addition, because $2(k-i)+1$ decreases during the trip, $r \leq \frac{d^+(s,t,k)}{d(s,t)}$ once the traveller selects the first strategy, i.e., following a path derived by GA.

Lemma 3 considers the competitive ratio of the GR algorithm under the condition $r > 2(k-i)+1$, in which the traveller selects the last two routing policies. Next, we combine the scenario with the first strategy, and Theorem 1 follows.

Lemma 3 *If the ratio* $r = \frac{d^+(v_i,t,k-i)}{d_{E_i}(s,t)} > 2(k-i)+1$, *for each* i *during the whole trip, then the total distance cost of* GR *satisfies the following property, where* $1 \leq i \leq k$, *provided there is a set of jammed edges* E_i .

$$d_{E_i^{GR}}(s,t) \leq \begin{cases} (i+1) \cdot d_{E_i}(s,t), & \textit{if the traveller uses } GA \textit{ at } v_i; \\ (2i+1) \cdot d_{E_i}(s,t), & \textit{if the traveller uses } RA \textit{ at } v_i. \end{cases}$$

Proof By induction on the number of jammed edges learned by the traveller, the proof is trivial for the case $i = 0$. Consider $i = 1$. The traveller walks from v_1 to t using GA when $d(s,t) + d_{E_1}(v_1,t) \leq (1+1) \cdot d_{E_1}(s,t)$. Thus, the travel cost is $d_{E_1^{GR}}(s,t) \leq d(s,t) + d_{E_1}(v_1,t) \leq 2d_{E_1}(s,t)$ for the case $i = 1$. Otherwise, the traveller uses RA at v_1; that is, the traveller returns to s, and takes the path with the cost $d_{E_1}(s,t)$. Thus, by Eq. (1), the total cost is at most $2d(s,t) + d_{E_1}(s,t) \leq (2+1) \cdot d_{E_1}(s,t)$.

Assume the statement holds for $i \leq \ell$. We divide the case $i = \ell + 1$ into two parts:

Case 1: The traveller chooses GA at $v_{\ell+1}$; that is, $d_{E_\ell^{GR}}(s,t) + d_{E_{\ell+1}}(v_{\ell+1},t) \le ((\ell+1)+1) \cdot d_{E_{\ell+1}}(s,t)$. Hence, the statement follows, because

$$d_{E_{\ell+1}^{GR}}(s,t) \le d_{E_\ell^{GR}}(s,t) + d_{E_{\ell+1}}(v_{\ell+1},t) \le (\ell+2) \cdot d_{E_{\ell+1}}(s,t).$$

Case 2: The traveller chooses RA at $v_{\ell+1}$. Consider two scenarios when the traveller was at v_ℓ. If the traveller used GA at v_ℓ, then $d_{E_\ell^{GR}}(s,t) \le (\ell + 1) \cdot d_{E_\ell}(s,t)$ by the induction hypothesis. Similarly, the traveller would go back to s from $v_{\ell+1}$ and take the path with the cost $d_{E_{\ell+1}}(s,t)$. Therefore, we have

$$\begin{aligned} d_{E_{\ell+1}^{GR}}(s,t) &\le 2 \cdot d_{E_\ell^{GR}}(s,t) + d_{E_{\ell+1}}(s,t) \le 2(\ell+1) \cdot d_{E_\ell}(s,t) + d_{E_{\ell+1}}(s,t) \\ &\le (2(\ell+1)+1) \cdot d_{E_{\ell+1}}(s,t) \end{aligned}$$

by Eq. (1). On the other hand, if the traveller used RA at v_ℓ, then by the induction hypothesis, $d_{E_\ell^{GR}}(s,t) \le (2\ell+1) \cdot d_{E_\ell}(s,t)$. Because the traveller started again from s and took the path with the cost $d_{E_\ell}(s,t)$ after learning about the jammed edge at v_ℓ, the distance cost of the path that the traveller used to return to s from $v_{\ell+1}$ will not be larger than $d_{E_\ell}(s,t)$. Thus, we have

$$\begin{aligned} d_{E_{\ell+1}^{GR}}(s,t) &\le d_{E_\ell^{GR}}(s,t) + d_{E_\ell}(s,t) + d_{E_{\ell+1}}(s,t) \\ &\le (2\ell+1) \cdot d_{E_\ell}(s,t) + d_{E_\ell}(s,t) + d_{E_{\ell+1}}(s,t) \\ &\le (2(\ell+1)+1) \cdot d_{E_{\ell+1}}(s,t). \end{aligned}$$

The proof is complete. □

The above lemma shows the competitive ratio without using the first strategy, i.e., following a path derived by GA, irrespective of whether the traveller finds a jammed edge. We remark that if the traveller does not know the bound k of the number of jammed edges, that is, r cannot be derived initially, then the last two routing policies of the GR algorithm can obtain a $(2k+1)$-competitive ratio by Lemma 3.

We prove that the GR algorithm is $\min\{r, 2k+1\}$-competitive if the number of jammed edges is bounded from below by a given constant k.

Theorem 1 *For the* Double- valued Graph *problem, the competitive ratio of* GR *is at most* $\min\{r, 2k+1\}$ *when the number of jammed edges is up to a given constant* k*, where* $r = \frac{d^+(s,t,k)}{d(s,t)}$ *initially, and* r *might decrease during the trip.*

Proof First, if the ratio $r = \frac{d^+(s,t,k)}{d(s,t)} \le 2k+1$ initially, the traveller follows a path from s to t derived by GA. The competitive ratio is $\frac{d^+(s,t,k)}{d_{E_k}(s,t)} \le \frac{d^+(s,t,k)}{d(s,t)} = r \le 2k+1$. Otherwise, the traveller follows GA or RA until he/she discovers a jammed edge. Assume that $r = \frac{d^+(v_i,t,k-i)}{d_{E_i}(s,t)} > 2(k-i)+1$ when the traveller learns about each jammed edge $e_i = (v_i, v_{i'})$ during the trip $1 \le i \le k$. That is, the traveller will never use the first strategy. Therefore, the competitive ratio is within $2k+1$ by Lemma 3.

On the other hand, if $r \leq 2(k-j)+1$ while learning about a jammed edge $e_j = (v_j, v_{j'})$, for some j, the traveller will follow a path from v_j to t derived by GA. The travel cost is at most $d_{E_{j-1}^{GR}}(s,t)$ before the traveller arrives at v_j. Then the total distance cost can be formulated as follows:

$$\begin{aligned} d_{E_j^{GR}}(s,t) &\leq d_{E_{j-1}^{GR}}(s,t) + d^+(v_j, t, k-j) \\ &\leq (2(j-1)+1) \cdot d_{E_{j-1}}(s,t) + (2(k-j)+1) \cdot d_{E_j}(s,t) \\ &\leq 2k \cdot d_{E_k}(s,t). \end{aligned}$$

The second inequality holds by Lemma 3. □

Regarding the time complexity analysis, the number of iterations in the while loop, i.e., the number of updates for the ratio r, is at most k. Each of the three routing strategies can apply Dijkstra's algorithm Dijkstra (1959) to devise a path from s or v_i to t, for some i. In addition, RA just takes the original s, v_i-path when the traveller needs to return to s from v_i. Thus, for a given constant k, the running time is a constant factor times $D(n)$, where n is the order of a graph G, and $D(n)$ is the running time of Dijkstra's algorithm.

The GR algorithm can also be extended to the Multiple- valued Graph problem in which each edge is associated with more than two possible distances. We regard the largest distance of each edge e as $d^+(e)$, and then $r = \frac{d^+(s,t,k)}{d(s,t)}$ is defined similarly. GR performs in a similar way to travel from s to t. Therefore, the competitive ratio remains the same.

Corollary 1 *For the* Multiple- valued Graph *problem, there is a* $\min\{r, 2k+1\}$*-competitive algorithm when the number of traffic jams is up to a given constant k.*

In addition, we consider the Recoverable k-CTP in which each blocked edge e is associated with a recovery time $r(e)$ to reopen. In this online problem, it is assumed that the blocked edges will not be blocked again. An instance I of the Recoverable k-CTP can be transformed into an instance I' of the Double- valued Graph problem by letting $r(e)$ be represented in terms of the distance, and letting $d^+(e) = d(e) + r(e)$ for every edge $e \in E$ in I'.

The GR algorithm can then be modified as follows. When learning about a jammed edge e_i, the traveller decides if he/she will traverse the edge e_i via GR. If the traveller traverses the jammed edge e_i, then $E_i = E_{i-1} \cup \{e_i\}$, and after passing through the edge e_i, e_i will be reset to its original distance $d(e_i)$. All the other operations are performed as described earlier, and the next corollary follows immediately.

Corollary 2 *For the Recoverable k-CTP, there is a* $\min\{r, 2k+1\}$*-competitive algorithm when the number of blockages is bounded from below by a given constant k.*

Su and Xu (2004) considered the Recoverable k-CTP and proposed two policies: a *greedy* strategy and a *waiting* strategy. They also investigated the problem and its variants in special road networks Su et al. (2008). When the number of blocked edges is up to a given constant k, Table 1 compares the competitive ratios of the previous

Table 1 Comparison of the GR results and the results reported in Su and Xu (2004), where $\alpha > 0$

Recovery time	Greedy strategy Su and Xu (2004)	Waiting strategy Su and Xu (2004)	GR
$r(e) \leq d(e)$	2^k	2	$r \leq 2$
$r(e) \leq \alpha \cdot d(e)$	$(1+\alpha)^k$	$1+\alpha$	$\min\{1+\alpha, 2k+1\}$
$r(e) \leq d(s,t)$	2^k	$k+1$	$r \leq k+1$
$r(e) \leq \alpha \cdot d(s,t)$	$(1+\alpha)^k$	$\alpha k+1$	$\min\{\alpha k+1, 2k+1\}$

strategies with the ratio of GR under different scenarios. For example, if $r(e) \leq \alpha \cdot d(e)$, $\alpha > 0$, i.e., $d^+(e) \leq (1+\alpha)d(e)$ in the Double- valued Graph problem, then $r = \frac{d^+(s,t,k)}{d(s,t)} \leq 1+\alpha$. If $\alpha \leq 2k$, GR will initially follow the path derived by GA. Thus, we have the competitive ratio $r \leq 1+\alpha$. Otherwise, if $\alpha > 2k$, the competitive ratio of GR is at most $2k+1$ by Lemma 3. Besides, if $r(e) \leq \alpha \cdot d(s,t)$, it implies that $r = \frac{d^+(s,t,k)}{d(s,t)} \leq \alpha k + 1$. Therefore, if $\alpha \leq 2$, GR will follow a path derived by GA, and we have the competitive ratio $r \leq \alpha k + 1$. Otherwise, if $r(e) > 2d(s,t)$, then $r > 2k+1$, and the competitive ratio of GR is at most $2k+1$ by Lemma 3. Note that the results show that the GR approach is at least as good as the previous results in Su and Xu (2004) when the number of blockages is bounded from below by a given constant k.

4 The uniform jam cost model

In this section, we study the uniform jam cost model. Suppose the jam cost of each edge e is a constant c, i.e., $d^+(e) = d(e) + c$. An online algorithm A is said to be a c^A-*additive competitive algorithm* for the Double- valued Graph problem if for any instances,

$$d_{E_i^A}(s,t) \leq d_{E_k}(s,t) + c^A, \quad 1 \leq i \leq k,$$

where c^A is a constant. We propose a tight lower bound below.

Lemma 4 *For the uniform jam cost model of the* Double- valued Graph *problem with at most k traffic jams, there is no deterministic online algorithm within a kc-additive competitive ratio; that is, given a uniform jam cost c, the derived solution cannot be better than* $d_{E_k}(s,t) + kc$.

Proof Consider the example in Fig. 2. In this graph, each edge e is associated with two possible distances: $d(e)$ and $d(e)+c$. There are $k+1$ different s, t-paths of length $k+1$, denoted by $P_1, \ldots, P_{k+1}$. For every edge $(s, v_{i,1})$, $1 \leq i \leq k+1$, $d(s, v_{i,1}) = \frac{1}{2}c$, and its jammed distance is $\frac{3}{2}c$. For each e of the other edges, $d(e) = \varepsilon$ and $d^+(e) = \varepsilon + c$.

For any deterministic algorithms, the traveller has three ways to walk from s to t. (1) The traveller simply heads for t directly, irrespective of whether he/she discovers a jammed edge; in this case, the travel cost is exactly $\frac{1}{2}c + k(\varepsilon + c)$. (2) The traveller is

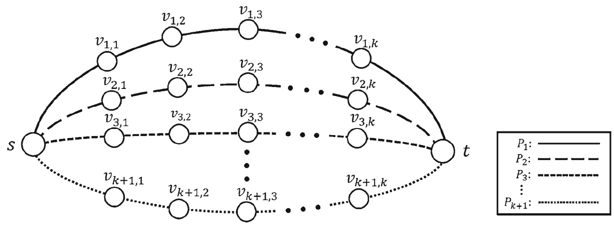

Fig. 2 An example of the lower bound in the uniform jam cost model

prevented from passing through jammed edges to reach t; that is, the traveller always goes back to s to avoid jams while learning about a jammed edge. Thus, the distance cost is at least $2(\frac{1}{2}c)k + \frac{1}{2}c + k\varepsilon$. (3) In the last case, the traveller might walk through several, but not all, jammed edges instead. From the malicious adversary point of view, it would be better not to assign jams to $(s, v_{i,1})$, for any i. When the traveller takes a path P_j, the malicious adversary will assign jams to $(v_{j,1}, v_{j,2})$, $(v_{j,2}, v_{j,3})$, ..., and so on, until the traveller returns to s. Thus, if the traveller traverses i paths, and returns to s in each iteration, the current cost is at least $2(\frac{1}{2}c)i + 2(\varepsilon + c)\ell$ while passing through ℓ jammed edges. There are still $k - (i + \ell)$ unknown jammed edges. Next, the traveller is prevented from passing through jammed edges to reach t. The total travel cost is:

$$
\begin{aligned}
&(ci + 2\varepsilon\ell + 2c\ell) + 2(\frac{1}{2}c)(k - (i + \ell)) + (\frac{1}{2}c + k\varepsilon)\\
=&(\frac{1}{2}c + k\varepsilon) + kc + (2\varepsilon + c)\ell \geq (\frac{1}{2}c + k\varepsilon) + kc.
\end{aligned}
$$

Thus, compared with the offline optimum $\frac{1}{2}c + k\varepsilon$, the additive competitive ratio is at least kc. □

By Lemma 4, no deterministic algorithm can derive a better additive competitive ratio than kc. Actually, the traveller can use a very straightforward algorithm to achieve the lower bound: following the s, t-path with the cost $d(s, t)$, irrespective of whether the traveller finds jammed edges. However, it is possible to slightly improve the ratio under some conditions.

For instance, if $c \geq 2\delta \cdot d(s, t)$ for some $\delta > 1$, we let a threshold be $\frac{c}{2\epsilon}$, for a constant $1 < \epsilon < \delta$. Assume there is a nonempty subset of jammed edges E_i, such that $d_{E_i}(s, t) \leq \frac{c}{2\epsilon}$ for some $i \leq k$, when the traveller learns about jammed edges. Based on this assumption, when $d_{E_j}(s, t) \leq \frac{c}{2\epsilon}$, the traveller will use RA; however, when $d_{E_j}(s, t) > \frac{c}{2\epsilon}$, the traveller will use GA until he/she arrives at t. Thus, we let e_ℓ be the first jammed edge, such that $d_{E_\ell}(s, t) > \frac{c}{2\epsilon}$, if any. That is,

$$
\begin{cases}
d_{E_j}(s, t) \leq \frac{c}{2\epsilon}, & \text{if } 1 \leq j < \ell; \qquad \triangleright \text{ the traveller uses } RA\\
d_{E_j}(s, t) > \frac{c}{2\epsilon}, & \text{if } \ell \leq j \leq k. \qquad \triangleright \text{ the traveller uses } GA
\end{cases}
$$

Then, the total travel cost can be formulated as follows:

$$\begin{aligned}
&2 \cdot d(s,t) + \ldots + 2 \cdot d_{E_{\ell-2}}(s,t) + d_{E_{\ell-1}}(s,t) + (k-\ell+1)c \\
&\le 2(\ell-1) \cdot \frac{c}{2\epsilon} + d_{E_{\ell-1}}(s,t) + (k-\ell+1)c \\
&\le d_{E_k}(s,t) + kc - (\ell-1)(1-\frac{1}{\epsilon})c.
\end{aligned}$$

If there is no such ℓ, the traveller uses RA until he/she arrives at t. The above equation implies that the total travel cost is at most $2k(\frac{c}{2\epsilon}) + d_{E_k}(s,t) = \frac{kc}{\epsilon} + d_{E_k}(s,t)$.

5 Concluding remarks

In this paper, we have studied the Double- valued Graph problem, which is a generalization of k-CTP, when the number of traffic jams is up to a given constant k. We have presented tight lower bounds and an adaptive algorithm that can satisfy the lower bound. In addition, we have extended the algorithm to the Recoverable k-CTP. We have also derived a lower bound with an additive competitive ratio for the uniform jam cost model.

It would be worthwhile investigating these online route planning problems because they find real applications in dynamic navigation systems designed to avoid traffic congestion. We conclude the study with two observations: First, compared with the *larger* lower bound of deterministic algorithms, it would be very interesting to develop a randomized online algorithm that can yield a better competitive ratio. On the other hand, a traveller could learn of a blockage or traffic congestion in advance from road sensor networks; for example, a GPS navigation system could indicate traffic conditions as the traveller approaches within a distance ℓ of an end vertex of a blocked edge (or a jammed edge) for a given constant ℓ. The question is how much the earlier information could improve online route planning. We will consider these issues in our future research.

Acknowledgements This work was supported by the National Science Council of Taiwan under Grants NSC100-2221-E-007-108-MY3 and NSC100-3113-P-002-012, and the Advanced Manufacturing and Service Management Research Center, National Tsing Hua University under Toward World-Class Universities Projects 100N2074E1 and 101N2074E1.

References

Bar-Noy A, Schieber B (1991) The Canadian traveller problem. Proceedings of the 2nd ACM-SIAM symposium on discrete algorithms (SODA), pp. 261–270

Ben-David S, Borodin A (1994) A new measure for the study of online algorithms. Algorithmica 11(1): 73–91

Borodin A, El-Yaniv R (1998) Online computation and competitive analysis. Cambridge University Press, Cambridge

Dijkstra EW (1959) A note on two problems in connexion with graphs. Numeri Math 1(1):269–271

Karger D, Nikolova E (2008) Exact algorithms for the Canadian traveller problem on paths and trees. Technical report, MIT Computer Science and Artificial Intelligence Lab. URL http://hdl.handle.net/1721.1/40093

Papadimitriou CH, Yannakakis M (1991) Shortest paths without a map. Theor Comput Sci 84(1):127–150
Sleator D, Tarjan RE (1985) Amortized efficiency of list update and paging rules. Commun ACM 28: 202–208
Su B, Xu YF (2004) Online recoverable Canadian traveller problem. Proceedings of the international conference on management science and engineering, pp. 633–639
Su B, Xu YF, Xiao P, Tian L (2008) A risk-reward competitive analysis for the recoverable Canadian traveller problem. Proceedings of the 2nd conference on combinatorial optimization and applications (COCOA). LNCS 5165:417–426
Westphal S (2008) A note on the k-Canadian traveller problem. Inf Process Lett 106:87–89
Xu YF, Hu ML, Su B, Zhu BH, Zhu ZJ (2009) The Canadian traveller problem and its competitive analysis. J Comb Optim 18:195–205