Theoretical Computer Science 530 (2014) 80–88

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The Covering Canadian Traveller Problem [☆]

Chung-Shou Liao [*,1], Yamming Huang

Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan

ARTICLE INFO

Article history:
Received 3 September 2013
Received in revised form 28 January 2014
Accepted 20 February 2014
Communicated by P.G. Spirakis

Keywords:
Canadian traveller problem
Competitive ratio
Online algorithm
Travelling salesman problem

ABSTRACT

The Canadian Traveller Problem (CTP) is to find the shortest route from a source to a destination under uncertain conditions. Some roads may be blocked by accidents, and a traveller only discovers a blockage upon reaching an adjacent site of the road. More precisely, the objective of this problem is to design an adaptive strategy against a malicious adversary who sets up road blockages in order to maximize the gap between the distance cost resulting from the strategy and the distance cost of the static shortest path in which all blockages are known a priori. This study investigates a variation of the Travelling Salesman Problem that involves finding the shortest tour, visiting a set of locations, and returning to the origin under the same uncertainty as that of the CTP. This online routing problem, called the Covering Canadian Traveller Problem (CCTP), has real applications in dynamic navigation systems such as delivery routing in express logistics. We study this problem from a competitive analysis perspective, and present an efficient touring strategy within an $O(\sqrt{k})$-competitive ratio, where the number of blockages is up to k. We also demonstrate the tightness of the competitive analysis.

1. Introduction

The need for *online* route planning in road networks is the primary motivation for this study. Although current navigation systems use information about road distances and speed limits to find the fastest routes, traffic conditions are often unpredictable and vary enormously. Therefore, online decision-making strategies play an important role in solving traffic congestion problems.

The *Canadian Traveller Problem* (CTP), defined by Papadimitriou and Yannakakis in 1991 [20], is to find the shortest path between a source and a destination, given incomplete information that is acquired in a dynamic manner. Consider a road network $G = (V, E)$ represented by a set V of transport stops connected by roads, where each road $e \in E$ is associated with a distance cost when a traveller traverses it. The traveller is aware of the entire structure of the network G in advance. However, some roads may be blocked by accidents, and a blockage becomes evident only when the traveller reaches a stop that is adjacent to the blocked road. Fig. 1 illustrates the route implementation of a traveller in a road network. The traveller initially follows the planned route to the destination t. After learning about an accident on a road during the trip, the traveller adjusts the route according to this *dynamic change*. This example shows how unpredictable accidents significantly affect planned routes, increasing the complexity of the CTP.

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

* Corresponding author.
E-mail addresses: csliao@ie.nthu.edu.tw (C.-S. Liao), s100034528@m100.nthu.edu.tw (Y. Huang).

1 This work was supported by the National Science Council of Taiwan under Grants NSC100-2221-E-007-108-MY3 and NSC102-2221-E-007-075-MY3.

http://dx.doi.org/10.1016/j.tcs.2014.02.026
0304-3975/

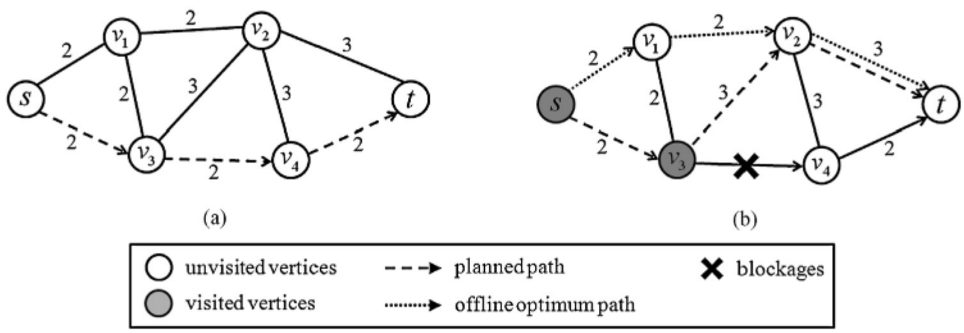

Fig. 1. An example that illustrates an unpredictable blockage occurring in a planned route.

The objective of this famous online routing problem is to plan an efficient route between a source and a destination under this uncertain scenario such that the competitive ratio is minimized. The competitive ratio represents the worst-case ratio of the distance cost of the planned route to the distance cost of the *static* shortest path in hindsight, where all blockages are known a priori. The next section provides a formal definition of the performance measure adopted in this study.

Dynamic optimization problems that involve finding the shortest route in an online fashion have been widely studied for decades. For example, Thomas and White [28,29] investigated stochastic models of the vehicle routing problem and the shortest path problem, respectively, where certain roads may be congested with known probabilities. This simplifying assumption provided new insights into the stochastic shortest path problem [22,23]. In addition, modern technologies such as Global Positioning Systems (GPS) and mobile communication have contributed to the development of dynamic navigation-planning based on real-time information. Häme and Hakula [12] explored the *dynamic* stochastic shortest path problem in which a transfer between two transport services might be rendered unsuccessful. In this case, the remaining routes to the destination must be reconsidered. This problem is similar to the stochastic model of the CTP, in which an independent blockage probability for each road is given. Bar-Noy and Schieber [5] and Karger and Nikolova [16] solved this stochastic CTP by adopting techniques from the Markov Decision Process. Recent reviews by Pillac et al. [21] and Larsen et al. [19] provided related research on these real-time route planning problems, which may be referred to as *dynamic* or *online* in the literature.

The *online* shortest path problem has also been studied from a different viewpoint [4,11,17]. A traveller makes a sequence of decisions to choose a route from a source to a sink against an adaptive adversary such that the cumulative loss is minimized over a sufficiently long period of time. The objective of this *online decision learning* problem (i.e., a repeated two-player game in an adversarial setting) is to find favorable routes in a series of trials, and then compare them with the best single decision made with the benefit of hindsight. The *multi-armed bandit* model allows the traveller to learn about the losses of only the routes that have been chosen, which is somewhat similar to the assumption of the CTP. However, the CTP is a single-round, two-player game in which the traveller only learns about traffic congestion upon reaching an adjacent site. In other words, the traveller cannot derive useful information using a chain of decisions. Therefore, the major difficulty in designing a favorable strategy under these conditions is to make decisions without any predictable traffic information regarding future blockages.

This study presents a generalization of the CTP, called the *Covering CTP* (CCTP) in an attempt to develop an efficient tour for a traveller visiting a set of locations and returning to the origin under the same uncertainty as that of the CTP. The CCTP is a variation of the *Travelling Salesman Problem* (TSP). To analyze the performance of online algorithms for this routing problem, we refer to previous research [5,32] and make two basic assumptions. First, the given road network remains connected even if the blocked roads are removed. That is, the traveller can reach every location even if the blocked roads are eliminated. Second, the state of a road does not change after the traveller learns about the blockage on the road; that is, once a road is blocked by an accident, the traveller can never pass through that road. This problem finds practical applications in dynamic navigation systems designed to avoid traffic congestion, and it is motivated by the *dynamic TSP* [19,22] and the *online TSP* [2,3,14,15]. The dynamic TSP has been investigated for various degrees of dynamism, such as changing pairwise distances between locations and adding or deleting locations [18,30]. Researchers have also proposed a number of evolutionary approaches for solving this problem [19,21]. On the other hand, Ausiello et al. [3] introduced the online TSP in which the set of locations that must be visited is revealed in a dynamic manner; more precisely, a sequence of requests for visiting locations in a metric space arrives in an online fashion. Ascheuer et al. [2] and Jaillet and Wagner [14] considered more general cases and proposed several polynomial time algorithms with constant competitive ratios. These studies that investigated the online routing problem were conducted based on the concept of competitive analysis [15]. Other previous research [1,10,24] provided more comprehensive information on static and deterministic TSP.

The main contribution of this study is summarized as follows. We consider the CCTP and propose an online touring algorithm within an $O(\sqrt{k})$-competitive ratio when the number of blockages is up to k. In addition, we present a tight example that demonstrates the worst-case ratio. The remainder of this paper is organized as follows. In Section 2, we introduce related research on the CTP and some definitions and notations. We present the main strategy for this online routing problem in Section 3. We analyze the competitive ratio of the proposed algorithm and show that the analysis is tight in Section 4. Finally, we conclude the paper with suggestions for future work and open questions.

2. Preliminaries

Papadimitriou and Yannakakis presented a PSPACE-completeness proof for the CTP when determining if there is an online strategy that guarantees a bounded competitive ratio [20]. They also showed that its stochastic model is #P-hard when considering the expected competitive ratio to the offline optimum path. In past decades, researchers have presented no substantial improvement in approximation algorithms for solving the CTP.

2.1. Past work on CTP

Because of the PSPACE-completeness result, subsequent studies have primarily focused on the k-CTP in which the number of blockages is up to k. Bar-Noy and Schieber [5] first proved that for an arbitrary k, the problem of devising an algorithm that guarantees a given travel distance remains PSPACE-complete. They investigated several variants of the CTP from a worst-case scenario viewpoint. In this case, the goal was to find a static (offline) strategy that minimizes the maximum possible travel cost. Bar-Noy and Schieber also proposed a recoverable model of the k-CTP in which each blocked edge is associated with a recovery time to reopen. Su and Xu [26] and Su et al. [27] considered the recoverable model and proposed a *greedy* approach and a *waiting* approach, respectively, for solving this problem.

Westphal presented important lower bounds for the k-CTP in 2008, which showed that no deterministic online algorithm can achieve a competitive ratio less than $(2k+1)$; moreover, any randomized algorithms cannot improve the deterministic lower bound substantially and cannot attain an expected competitive ratio smaller than $(k+1)$ [31]. In addition, this study proposed a simple *reposition* strategy that achieves the deterministic lower bound. This strategy requires a traveller to repeat the following: traverse a shortest path from the source to the destination until finding a blockage on the way, return to the source through the path, and select a new shortest path based on the updated blockage information. Xu et al. [32] presented a similar *comparison* strategy for the k-CTP. Their strategy incorporates the concept of reposition, and can attain the lower bound as well. They also used Dijkstra's algorithm [9] to discuss the performance of the *greedy* strategy, which always selects a shortest path to the destination based on the current blockage information, when learning about a blockage on the way. They showed that the competitive ratio of the greedy algorithm is exponential in k in the worst case. However, its performance is relatively acceptable in practical situations such as urban road networks. Recently, Bender and Westphal proposed the first randomized algorithm for special graphs in which all paths between a given source-sink pair are vertex-disjoint [6]. Huang and Liao considered a generalization of the k-CTP, called the *Double-valued Graph*, in which each edge is associated with two possible distances [13]. They proposed similar lower bounds and a polynomial time algorithm that meets the proposed lower bound.

2.2. Definitions and notations

The definition of the CCTP is as follows. Given a complete graph $G=(V,E)$ with an origin s, a traveller knows the entire structure of G and wishes to begin at s, visit every other vertex in V, and return to s as quickly as possible. However, the traveller discovers online that some edges are blocked once reaching them. Precisely, the traveller traverses the graph under the same uncertainty of the CTP; that is, the traveller only discovers a blocked edge upon reaching an end vertex adjacent to the blockage. Moreover, as mentioned earlier, we make two assumptions: one is that once a blocked edge is learned by the traveller, the edge remains blocked forever. The other is that the complete graph G remains connected even if the blocked edges are eliminated.

This study considers the k-CCTP with at most k blockages. Because a complete graph G is $(n-1)$-edge-connected, where n is the order of G (i.e., $|V|=n$), the above assumption constrains the number of blockages to be less than the edge connectivity of G, i.e., $k<n-1$. Let $E_i^A=\{e_1,e_2,\dots,e_i\}\subseteq E$, $1\leqslant i\leqslant k$, consist of the blocked edges that are learned by an online strategy A during the trip, and let E_k be the set of all blockages. In what follows, the superscript A can be ignored without causing confusion. In addition, let $d:E\to R^+$ be the distance cost function. We denote the travel cost from u to v as $d_{E_i^A}(u,v)$, derived by an adaptive algorithm A based on blockage information E_i that is learned during the trip. Furthermore, let $d_{E_k}(s,s)$ be the static (offline) optimum (an optimal cycle from s through every other vertex to s) of the k-CCTP under full blockage information E_k. The next property immediately follows [13,32]:

$$d(s,s)\leqslant d_{E_1}(s,s)\leqslant\cdots\leqslant d_{E_k}(s,s),\quad \text{where } E_1\subseteq E_2\subseteq\cdots\subseteq E_k. \tag{1}$$

For convenience, we denote a path p from u to v as $p:u\sim v$, and a cycle tour P of length ℓ as $P:s=v_1-v_2-\cdots-v_\ell-s$, where P has ℓ edges.

Competitive analysis is the most widely accepted method of measuring the performance of adaptive strategies for an online routing problem [7,15,25]. The competitive ratio of an online algorithm A can be formally defined as follows [7,25]: for any instances, the competitive ratio is defined as the worst case ratio of the online algorithm A's total cost to the cost of the optimal offline algorithm. Therefore, it is said that an online algorithm A is c^A-*competitive* for the k-CCTP, if for any instances,

Algorithm 1: Cyclic Routing (CR).

Input: $G = (V, E)$ of order n with an origin s and $d : E \to R^+$;
Output: A tour P^{cr} that visits every vertex in V;

Use the Christofides algorithm to find a tour $P : s = v_1 - v_2 - \cdots - v_n - s$ that visits every vertex exactly once, and initialize $m = 1$, $V_1 \leftarrow V \setminus \{s\}$, and $v_{1,0} \leftarrow s$;
while $V_m \neq \emptyset$ **do**
  Let the shortcut of P that traverses through V_m and follows the order of P be $P_m : v_{m,0} - v_{m,1} - \cdots - v_{m,|V_m|}$;
  if $m = 1$ or $v_{m,0} = v_{m-1,|V_{m-1}|}$ **then**
    Perform the SHORTCUT procedure in the same direction as that in the $(m-1)$th round or in the visiting order of P when $m = 1$;
    if $V_{m+1} = V_m$ **then**
      Perform the SHORTCUT procedure in the opposite direction;
    end if
  else
    Perform the SHORTCUT procedure in the opposite direction to that in the $(m-1)$th round;
  end if
  $m \leftarrow m + 1$;
end while
The traveller returns to s directly or finds a previously visited vertex u such that $v_{m,0} - u - s$ is not blocked and traverses the alternative path;
Concatenate the two paths P^{cr} and $v_{m,0} \sim s$;
return P^{cr};

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

where ε is a constant. This paper investigates the k-CCTP with the goal of designing an online routing strategy that minimizes its competitive ratio while enabling the traveller to visit every vertex and return to the origin under the same uncertainty of the k-CTP.

3. Main algorithm

We note that this online routing problem, CCTP allows a traveller to visit a location more than once because blockages may necessitate this. We also assume that the distance costs satisfy the *triangle inequality*, so that the well-known Christofides algorithm [8] provides a 1.5-approximation solution for the *metric* TSP as an initial planned tour.

The rationale behind the proposed Cyclic Routing (CR) algorithm (see Algorithm 1) is as follows: the entire route can be decomposed into several rounds; in each round, the traveller attempts to visit as many vertices as possible in a given complete graph G following the visiting order of the tour derived by the Christofides algorithm, denoted by P: $s = v_1 - v_2 - \cdots - v_n - s$. In order to do this, the CR algorithm explores unvisited vertices via shortcuts to the tour P while the traveller discovers blockages. More precisely, the algorithm follows the visiting order of P and traverses a shortcut to P through as many unvisited vertices as possible in each round. The reason is that the cost of such a shortcut is at most the cost of the tour P because of the triangle inequality; moreover, the cost of the tour P is actually a lower bound on the offline optimum of the k-CCTP, which will be clarified later in the next section. When the algorithm repeats this process until every vertex is visited, the number of rounds determines its competitive ratio. Because in each round the CR algorithm needs to perform a shortcut leaving some vertices unvisited due to newly discovered blockages, we claim that the number of rounds is bounded by a constant times the square root of the number of blockages. We will show the competitive analysis later.

Assume the entire route is decomposed into m^{cr} rounds, in each of which the traveller moves through the tour P. Let V_m be the set of unvisited vertices in the mth round, where $1 \leqslant m \leqslant m^{cr}$. Suppose $V_0 = V$ and $V_{m^{cr}+1} = \emptyset$ for the two dummy rounds $m = 0$ and $m = m^{cr} + 1$, respectively, and let $V_1 = V \setminus \{s\}$. Therefore, $V \supset V_1 \supseteq V_2 \supseteq \cdots \supseteq V_{m^{cr}} \supset V_{m^{cr}+1}$. In addition, we denote the shortcut of the tour P that traverses through V_m and follows the order of P as P_m: $v_{m,0} - v_{m,1} - \cdots - v_{m,|V_m|}$ when the traveller wants to visit the unvisited vertices in V_m in the mth round. When the traveller attempts to visit P_m over V_m, we denote the path derived by the SHORTCUT procedure as P_m^{cr}. Note that $v_{m,0}$ is the last vertex visited in the $(m-1)$th round when $m > 1$ (i.e., $v_{m,0} \notin V_m$ and $v_{1,0} = s$).

As mentioned previously, let E_i^{CR} consist of the blocked edges revealed by the CR algorithm, $1 \leqslant i \leqslant k$. We let the subset of E_i^{CR} the traveller learns about in the mth round be denoted by $E_{m,i}^{CR}$, $1 \leqslant m \leqslant m^{cr}$; that is, $E_i = \{e_1, \ldots, e_i\} = \bigcup_{m=1}^{m^{cr}} E_{m,i}^{CR}$. For simplicity, we use E'_m to represent $E_{m,i}^{CR}$ without causing confusion.

Fig. 2 shows an example of the CR algorithm. In this example, the traveller begins at the vertex v_1 and moves in a clockwise direction, where the given tour is P: $v_1 - v_2 - \cdots - v_{16} - v_1$. The solid lines represent the path P_1: $v_1 - v_2 - \cdots - v_{16}$ in the first round ($m = 1$). Dashed lines represent a bypass route derived by the SHORTCUT procedure when the traveller learns about a new blockage along P_1. The *gray* and *white* vertices represent the visited and unvisited vertices, respectively,

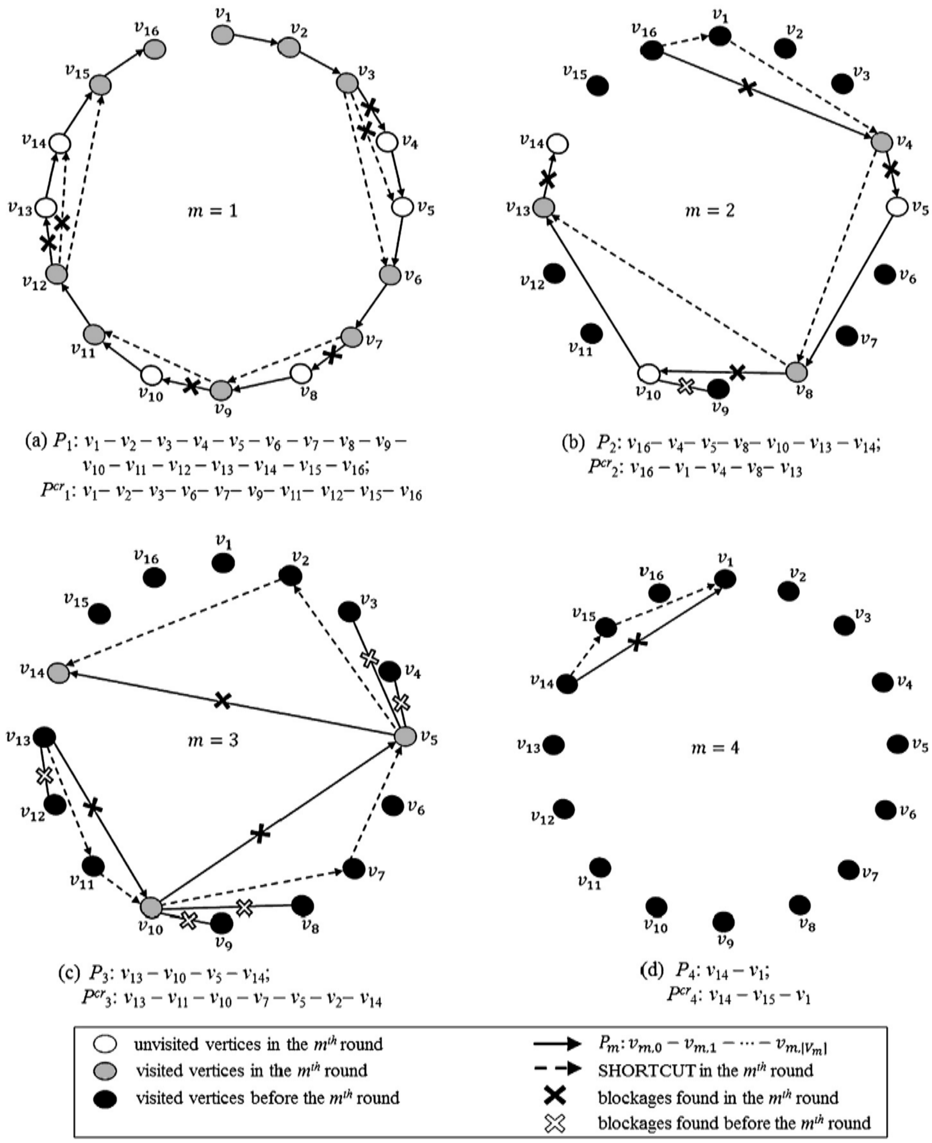

Fig. 2. An example of the process of the *CR* algorithm.

in this round. The trip includes six blockages, which are $E'_1 = \{(v_3, v_4), (v_3, v_5), (v_7, v_8), (v_9, v_{10}), (v_{12}, v_{13}), (v_{12}, v_{14})\}$. Therefore, six unvisited vertices v_4, v_5, v_8, v_{10}, v_{13}, and v_{14} exist after performing the SHORTCUT procedure; that is, $V_2 = \{v_4, v_5, v_8, v_{10}, v_{13}, v_{14}\}$ and the visited vertices become *dark* in the next round. Next, consider the second round (i.e., $m = 2$). The traveller begins at the vertex v_{16} and attempts to visit V_2 in the same direction as that in the first round because $v_{2,0} = v_{16} = v_{1,|V_1|}$. Following the *CR* algorithm, the traveller also traverses along the solid lines, and if necessary, takes bypass roads through internal visited vertices (i.e., the dashed lines based on the SHORTCUT procedure). The traveller learns about four blockages in this round: $E'_2 = \{(v_{16}, v_4), (v_4, v_5), (v_8, v_{10}), (v_{13}, v_{14})\}$. Therefore, the scenario produces $V_3 = \{v_5, v_{10}, v_{14}\}$; that is, only the gray vertices v_4, v_8, and v_{13} can be visited in the second round. Note that when the traveller moves from v_8 for v_{10}, the SHORTCUT procedure does not use the internal visited vertex v_9 because the blocked edge (v_9, v_{10}) was revealed in the previous round. Consider the third round. Because $v_{3,0} = v_{13} \neq v_{2,|V_2|} = v_{14}$, the traveller begins at the vertex v_{13} and visits V_3 in a counterclockwise direction. The traveller learns about three blocked edges (v_{13}, v_{10}), (v_{10}, v_5) and (v_5, v_{14}), and all the remaining gray vertices are visited in the third round (i.e., $V_4 = \emptyset$). The traveller stops at the vertex v_{14} and must return to $s = v_1$ in the last step. Finally, the traveller learns about a blocked edge (v_{14}, v_1) and traverses the dashed lines back to the origin v_1 following the SHORTCUT procedure. Note that in the mth round, $1 \leqslant m \leqslant m^{cr}$, the traveller may traverse the path P_m in either a clockwise direction or a counterclockwise direction, as mentioned above.

procedure SHORTCUT
  $V_{m+1} \leftarrow V_m$, $E'_m \leftarrow \emptyset$, $i = 0$ and $j = 1$;
  while $j \leqslant |V_m|$ **do**
    if $(v_{m,i}, v_{m,j})$ is not blocked **then** ▷ Visit $v_{m,j}$ in the mth round.
      $V_{m+1} \leftarrow V_{m+1} \setminus \{v_{m,j}\}$;
      Concatenate the path P^{cr} and the edge $(v_{m,i}, v_{m,j})$;
      $i \leftarrow j$ and $j \leftarrow i + 1$;
    else ▷ $(v_{m,i}, v_{m,j})$ is a blockage.
      $E'_m \leftarrow E'_m \cup \{(v_{m,i}, v_{m,j})\}$ and let v_ℓ be $v_{(m,i)+1}$;
      while $v_\ell \neq v_{m,j}$ **and** $v_{m,i} - v_\ell - v_{m,j}$ is blocked **do**
        $E'_m \leftarrow E'_m \cup \{e\}$, for the new blockage $e = (v_{m,i}, v_\ell)$ or $e = (v_\ell, v_{m,j})$;
        $\ell \leftarrow \ell + 1$;
      end while
      if $v_\ell \neq v_{m,j}$ **then** ▷ Visit $v_{m,j}$ via the internal visited vertex v_ℓ.
        $V_{m+1} \leftarrow V_{m+1} \setminus \{v_{m,j}\}$;
        Concatenate the two paths P^{cr} and $v_{m,i} - v_\ell - v_{m,j}$;
        $i \leftarrow j$ and $j \leftarrow i + 1$;
      else ▷ Skip $v_{m,j}$ in the mth round.
        $j \leftarrow j + 1$;
      end if
    end if
  end while
end procedure

4. Competitive analysis

In this section, we prove the competitive ratio of the *CR* algorithm, and show that the competitive analysis is tight.

Lemma 4.1. *The traveller can visit at least one vertex in V_m using CR in the mth round, $1 \leqslant m \leqslant m^{cr}$; that is, $|V_1| > |V_2| > \cdots > |V_m| > |V_{m+1}| > \cdots > |V_{m^{cr}}|$.*

Proof. Assume the traveller cannot visit any vertex in V_m for some m. The traveller starts at $v_{m,0}$. Consider the case in which $m = 1$ or $v_{m,0} = v_{m-1,|V_{m-1}|}$. Because the traveller does not use the SHORTCUT procedure to visit any vertices in V_m, a blockage exists for each edge $(v_{m,0}, v_{m,i})$, $1 \leqslant i \leqslant |V_m|$. A blocked edge also occurs on every alternative route $v_{m,0} - u - v_{m,i}$, $u \in V \setminus V_m$. The SHORTCUT procedure also checks if a shortcut $v_{m,0} - u - v_{m,|V_m|}$ exists in the opposite direction. Therefore, the number of blockages found is at least $|V_m| + (n - |V_m| - 1) = n - 1 > k$, which is a contradiction.

Otherwise, $v_{m,0} \neq v_{m-1,|V_{m-1}|}$. Then, we have $v_{m-1,|V_{m-1}|} \in V_m$ because $v_{m,0}$ is the last visited vertex in the $(m-1)$th round. The SHORTCUT procedure finds a blockage for every shortcut $v_{m,0} - u - v_{m-1,|V_{m-1}|}$ in the $(m-1)$th round. The traveller traverses the path P_m in the opposite direction to that in the $(m-1)$th round. Similarly, the procedure scans all the vertices from $v_{m,0}$ to at least $v_{m-1,|V_{m-1}|}$ to visit vertices in V_m directly or through alternative routes. Therefore, the number of blockages found is also at least $n - 1 > k$. This contradicts the assumption. The proof is complete. □

Based on this key lemma, the tour P^{cr} derived by the *CR* algorithm traverses all the vertices in V until the last round; that is, P^{cr} is a feasible solution. The next lemma states that the *CR* algorithm never rediscovers blockages, and this property is important for the competitive analysis.

Lemma 4.2. *$E'_i \cap E'_j = \emptyset$ for any $1 \leqslant i < j \leqslant m^{cr}$.*

Proof. Consider the path P_j over the set of unvisited vertices V_j, $1 < j \leqslant m^{cr}$. Obviously, each new blockage has both of its end vertices in V_j. In addition, if a blocked edge $e \in E'_i$, $1 \leqslant i < j$, was discovered in the ith round and one of its end vertices is in V_j, then the other end vertex is in $V_i \setminus V_j$; otherwise, it would be impossible for the traveller to find the blockage e whose two end vertices are unvisited. Therefore, a blockage in E'_j does not appear in E'_i. □

Lemma 4.3. *The number of new blockages discovered in the mth round is not less than the number of unvisited vertices in the $(m+1)$th round, i.e., $|E'_m| \geqslant |V_{m+1}|$, $1 \leqslant m \leqslant m^{cr}$.*

Proof. For every vertex v that remained unvisited in the $(m+1)$th round, there must be a newly visited u preceding it on the path such that (u, v) is a blocked edge. That is, for all such v there is a blockage discovered in the mth round. Therefore, when a blocked edge was discovered in E'_m, at most one vertex in V_m remains unvisited and it is in V_{m+1}. So $|E'_m| \geqslant |V_{m+1}|$, $1 \leqslant m \leqslant m^{cr}$. □

We developed the *CR* strategy based on a tour P derived by the Christofides algorithm for the original metric TSP. The Christofides algorithm is a combination of the minimum spanning tree of a complete graph G with the minimum weight perfect matching on the vertices with odd degree in the tree. The result of this algorithm is a Hamiltonian tour with a 1.5-approximation ratio if the distance function satisfies the triangle inequality property.

Let *OPT* be the offline optimum of the k-CCTP. Because the offline optimal tour visits every vertex and returns to the origin under full blockage information, the tour can be converted to a feasible solution to the original metric TSP through shortcuts. Thus, *OPT* is not smaller than the optimum of the metric TSP, denoted by OPT_{TSP}, because of the triangle inequality.

Lemma 4.4. *When the traveller attempts to traverse P_m using CR in the mth round, $1 \leqslant m \leqslant m^{cr}$, the travel cost is not larger than $3OPT$. In addition, when the traveller returns to the origin s in the final step, the travel cost is at most OPT.*

Proof. Let $d(P)$ denote the travel cost of the tour P, i.e., $d(P) = \sum_{i=1}^{n} d(v_i, v_{i+1})$, where $v_1 = v_{n+1} = s$. Obviously, $d(P) \geqslant d(P_m) = \sum_{i=1}^{|V_m|} d(v_{m,i-1}, v_{m,i})$ and $d(P_m^{cr}) \geqslant d(P_m)$, $1 \leqslant m \leqslant m^{cr}$. In each round, the traveller attempts to traverse the path P_m following the order of P or the opposite direction (see Line 5 to Line 10 in the *CR* algorithm). Thus, the traveller traverses the tour P at most twice in each round, and the travel cost is

$$d\big(P_m^{cr}\big) \leqslant 2d(P) \leqslant 2 \times 1.5 \times OPT_{TSP} \leqslant 3OPT,$$

because of the triangle inequality. That is, the travel cost is at most $3OPT$ in a single round. Consider the final step. The traveller can either traverse the edge $(v_{m+1,0}, s)$, or stop by a visited vertex, say u, and then return to s, i.e., $v_{m+1,0} - u - s$. Therefore, the travel cost is at most *OPT* because of the triangle inequality. □

Theorem 4.5. *The k-CCTP can be approximated within an $O(\sqrt{k})$-competitive ratio, when the number of blockages is up to k.*

Proof. By Lemma 4.2 and $|\bigcup_{m=1}^{m^{cr}} E'_m| \leqslant k$, we have

$$\big|E'_1\big| + \big|E'_2\big| + \cdots + \big|E'_{m^{cr}}\big| \leqslant k,$$

because any two E'_i and E'_j are disjoint, $1 \leqslant i < j \leqslant m^{cr}$. In addition, by Lemma 4.3, we have

$$|V_2| + |V_3| + \cdots + |V_{m^{cr}+1}| \leqslant \big|E'_1\big| + \big|E'_2\big| + \cdots + \big|E'_{m^{cr}}\big| \leqslant k. \tag{2}$$

In the worst case, $|V_m \setminus V_{m+1}| = 1$, $1 \leqslant m \leqslant m^{cr}$ by Lemma 4.1. That is, $|V_{m^{cr}+1}| = 0$, $|V_{m^{cr}}| = 1, \ldots,$ and $|V_2| = m^{cr} - 1$. By Eq. (2), the number of rounds is at most $O(\sqrt{k})$, which can be derived as follows:

$$\begin{aligned}
&\frac{(1 + (m^{cr} - 1))(m^{cr} - 1)}{2} \leqslant k \\
\Rightarrow\quad & \frac{1 - \sqrt{1 + 8k}}{2} \leqslant m^{cr} \leqslant \frac{1 + \sqrt{1 + 8k}}{2} \\
\Rightarrow\quad & m^{cr} \leqslant \left\lfloor \frac{1 + \sqrt{1 + 8k}}{2} \right\rfloor.
\end{aligned}$$

Let $d(P^{cr})$ denote the travel cost of the tour P^{cr} derived by *CR*. Then, by Lemma 4.4, the total cost of the tour is

$$d\big(P^{cr}\big) \leqslant \big(3m^{cr} + 1\big)OPT, \quad \text{where } m^{cr} = O(\sqrt{k}).$$

Therefore, the competitive ratio is within $O(\sqrt{k})$. □

Note that if the traveller does not know the upper bound k of the number of blockages (i.e., if k is unknown at the outset), the performance of the *CR* algorithm remains the same. In addition, we remark the *CR* algorithm takes polynomial time. The SHORTCUT procedure attempts to visit every vertex in V_m in each round. The procedure traverses the path P_m as well as the internal visited vertices between $v_{m,i}$ and $v_{m,j}$ if needed, $1 \leqslant i \leqslant j \leqslant |V_m|$. Moreover, the conditions in Line 4 and Line 10 in the SHORTCUT procedure can be verified in polynomial time because the number of blockages is up to k, where $k < n - 1$. Therefore, the procedure takes polynomial time in each round and so does the entire *CR* algorithm, because the number of rounds is at most $O(\sqrt{n})$.

The following corollary provides a tight example that attains the competitive ratio and thus shows the tightness of the analysis.

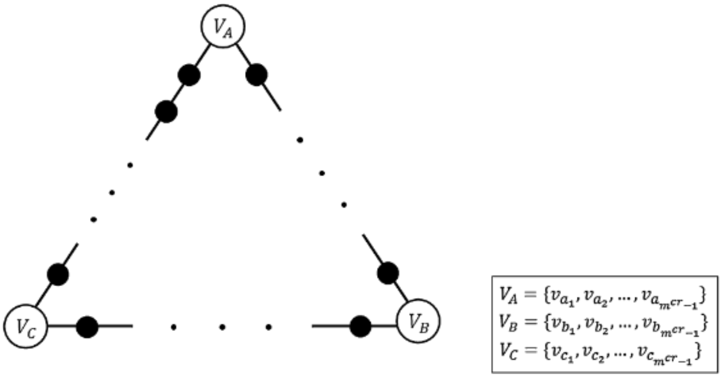

Fig. 3. A tight example of the $O(\sqrt{k})$-competitive ratio.

Corollary 4.6. *The competitive analysis of the $O(\sqrt{k})$-competitive CR algorithm is tight.*

Proof. We give a tight example to achieve the worst case bound asymptotically. Consider the graph in Fig. 3. The cycle represents the tour P derived by the Christofides algorithm, where V_A, V_B and V_C are three pseudo nodes, each of which contains $m^{cr} - 1$ vertices. The distance between any two vertices in a pseudo node is extremely small, whereas the distance between V_A, V_B, and V_C is significantly large. Let $V_A = \{v_{a_1}, v_{a_2}, \ldots, v_{a_{m^{cr}-1}}\}$, $V_B = \{v_{b_1}, v_{b_2}, \ldots, v_{b_{m^{cr}-1}}\}$, and $V_C = \{v_{c_1}, v_{c_2}, \ldots, v_{c_{m^{cr}-1}}\}$. In the first round, the traveller starts at an arbitrary vertex outside the pseudo nodes and moves in a clockwise direction. The traveller can visit all the vertices except those in the three pseudo nodes. Therefore, $|E'_1| = |V_2| = 3(m^{cr} - 1)$. Subsequently, in the $(i+1)$th round, $1 \leqslant i \leqslant m^{cr} - 1$, the traveller can visit only three vertices v_{a_i}, v_{b_i}, and v_{c_i} (i.e., only one vertex in each pseudo node). Therefore, $|E'_i| = |V_{i+1}| = 3(m^{cr} - i)$ and $V_{m^{cr}+1} = \emptyset$. Similar to the proof of Theorem 4.5, we have

$$\begin{aligned} &\left|E'_1\right| + \left|E'_2\right| + \cdots + \left|E'_{m^{cr}}\right| \\ &\quad = 3\left(m^{cr} - 1\right) + 3\left(m^{cr} - 2\right) + \cdots + 0 \leqslant k \\ &\quad \Rightarrow \quad m^{cr} \leqslant \left\lfloor \frac{1 + \sqrt{1 + \frac{8}{3}k}}{2} \right\rfloor. \end{aligned}$$

In each round, the traveller traverses the whole tour P, so the total travel cost is $d(P^{cr}) = O(m^{cr}) \times OPT$, where $m^{cr} = O(\sqrt{k})$. □

5. Concluding remarks

This study has presented the k-CCTP, which is a variation of the metric TSP under the same uncertainty as that of the k-CTP in which the number of blockages is up to k. We have proposed an $O(\sqrt{k})$-competitive algorithm for this online routing problem, and have also provided a tightness proof for this competitive analysis. In particular, the proposed algorithm can work and its competitive ratio remains the same even if the number of blockages k is unknown at the outset. Investigating the k-CTP and its variations is a worthy topic for future research because these problems are relevant to practical applications in dynamic route-planning systems, such as delivery routing in express logistics.

We conclude this study with two open problems as follows. First, given a road network, if multiple travellers can share real-time information about blockages when simultaneously exploring the road network, what is the problem complexity and approximation hardness? Zhang and Xu [33] recently discussed the k-CTP provided with two communication mechanisms. They presented several lower bounds and an adaptive strategy for special networks. When this scenario is applied to the k-CCTP, it would be of considerable interest to identify any appropriate strategies within a smaller competitive ratio with the help of multi-travellers and their communications. Unlike the online multi-vehicle routing problems reported in previous research [2,3,14,15], the different communications between travellers would affect the performance of online strategies for this problem. Second, the performance of the asymptotic competitive ratio of the proposed algorithm may be improved to reduce the constant factor, benefiting real implementations.

Acknowledgements

We wish to thank the anonymous reviewers for valuable comments and suggestions, which helped us improve the quality of the presentation of the paper.

References

[1] D.L. Applegate, R.E. Bixby, V. Chvátal, W.J. Cook, The Traveling Salesman Problem: A Computational Study, Princeton University Press, Princeton, New Jersey, 2006.
[2] N. Ascheuer, S. Krumke, J. Rambau, Online dial-a-ride problems: Minimizing the completion time, in: Proc. the 17th Int. Sympos. Theoret. Aspects Comput. Sci. (STACS), in: LNCS, vol. 1770, 2000, pp. 639–650.
[3] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, M. Talamo, Algorithms for the on-line travelling salesman, Algorithmica 29 (4) (2001) 560–581.
[4] B. Awerbuch, R. Kleinberg, Online linear optimization and adaptive routing, J. Comput. System Sci. 74 (2008) 97–114.
[5] A. Bar-Noy, B. Schieber, The Canadian traveller problem, in: Proc. the 2nd ACM-SIAM Symposium on Discrete Algorithms (SODA), 1991, pp. 261–270.
[6] M. Bender, S. Westphal, An optimal randomized online algorithm for the k-Canadian traveller problem on node-disjoint paths, J. Comb. Optim. (2013), Published online, http://dx.doi.org/10.1007/s10878-013-9634-8.
[7] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, Cambridge, 1998.
[8] N. Christofides, Worst-case analysis of a new heuristic for the traveling salesman problem, Technical report, Graduate School of Industrial Administration, Carnegie–Mellon University, 1976.
[9] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1) (1959) 269–271.
[10] G. Gutin, A.P. Punnen (Eds.), The Traveling Salesman Problem and Its Variations, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.
[11] A. György, T. Linder, G. Lugosi, G. Ottucsák, The on-line shortest path problem under partial monitoring, J. Mach. Learn. Res. 8 (2007) 2369–2403.
[12] L. Häme, H. Hakula, Dynamic journeying under uncertainty, European J. Oper. Res. 225 (3) (2013) 455–471.
[13] Y. Huang, C.S. Liao, The Canadian traveller problem revisited, in: Proc. 23rd International Symposium on Algorithms and Computation (ISAAC), in: LNCS, vol. 7676, 2012, pp. 352–361.
[14] P. Jaillet, M.R. Wagner, Generalized online routing: New competitive ratios, resource augmentation, and asymptotic analyses, Oper. Res. 56 (3) (2008) 745–757.
[15] P. Jaillet, M.R. Wagner, Online vehicle routing problems: A survey, in: B. Golden, S. Raghavan, E. Wasil (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces, vol. 43, Springer Science Publishers, 2008, pp. 221–237.
[16] D. Karger, E. Nikolova, Exact algorithms for the Canadian traveller problem on paths and trees, Technical report, MIT Computer Science & Artificial Intelligence Lab., 2008, URL http://hdl.handle.net/1721.1/40093.
[17] A. Kalai, S. Vempala, Efficient algorithms for online decision problems, J. Comput. System Sci. 71 (2005) 291–307.
[18] A. Larsen, O.B.G. Madsen, M.M. Solomon, The a priori dynamic traveling salesman problem with time windows, Transp. Sci. 38 (4) (2004) 459–472.
[19] A. Larsen, O.B.G. Madsen, M.M. Solomon, Recent developments in dynamic vehicle routing systems, in: B. Golden, S. Raghavan, E. Wasil (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces, vol. 43, Springer Science Publishers, 2008, pp. 199–218.
[20] C.H. Papadimitriou, M. Yannakakis, Shortest paths without a map, Theoret. Comput. Sci. 84 (1) (1991) 127–150.
[21] V. Pillac, M. Gendreau, C. Guéret, A.L. Medaglia, A review of dynamic vehicle routing problems, European J. Oper. Res. 225 (1) (2013) 1–11.
[22] H.E. Psaraftis, Dynamic vehicle routing problems, in: B. Golden, A. Assas (Eds.), Vehicle Routing: Methods and Studies, Elsevier Science Publishers, 1988, pp. 223–248.
[23] H.E. Psaraftis, J.N. Tsitsiklis, Dynamic shortest paths in acyclic networks with markovian arc costs, Oper. Res. 41 (1) (1993) 91–101.
[24] A. Schrijver, The traveling salesman problem, in: Combinatorial Optimization: Polyhedra and Efficiency, vol. B, Springer, Berlin, 2003, pp. 981–1004.
[25] D. Sleator, R.E. Tarjan, Amortized efficiency of list update and paging rules, Commun. ACM 28 (1985) 202–208.
[26] B. Su, Y.F. Xu, Online recoverable Canadian traveller problem, in: Proc. the International Conference on Management Science and Engineering, 2004, pp. 633–639.
[27] B. Su, Y.F. Xu, P. Xiao, L. Tian, A risk-reward competitive analysis for the recoverable Canadian traveller problem, in: Proc. the 2nd Conference on Combinatorial Optimization and Applications (COCOA), in: LNCS, vol. 5165, 2008, pp. 417–426.
[28] B.W. Thomas, Chelsea C. White III, Anticipatory route selection, Transp. Sci. 38 (4) (2004) 473–487.
[29] B.W. Thomas, Chelsea C. White III, The dynamic shortest path problem with anticipation, European J. Oper. Res. 176 (2007) 836–854.
[30] A. Toriello, W.B. Haskell, M. Poremba, A dynamic traveling salesman problem with stochastic arc costs, in: Proc. 13th INFORMS Computing Society Conference (ICS), 2013.
[31] S. Westphal, A note on the k-Canadian traveller problem, Inform. Process. Lett. 106 (2008) 87–89.
[32] Y.F. Xu, M.L. Hu, B. Su, B.H. Zhu, Z.J. Zhu, The Canadian traveller problem and its competitive analysis, J. Comb. Optim. 18 (2009) 195–205.
[33] H. Zhang, Y.F. Xu, The k-Canadian traveller problem with communication, in: Proc. the 5th International Frontiers of Algorithmics Workshop (FAW-AAIM), in: LNCS, vol. 6681, 2011, pp. 17–28.