Journal of Combinatorial Optimization (2022) 44:1039–1059
https://doi.org/10.1007/s10878-022-00876-9

Online routing and searching on graphs with blocked edges

Davood Shiri[1] · **Hakan Tozan**[2]

Accepted: 11 June 2022 / Published online: 22 June 2022

Abstract
We study routing and searching decisions of a search-and-detection (SDT) team on a road network under online uncertainty setting. Given an undirected edge-weighted bounded graph, a static target is positioned at an unknown vertex among potential target vertices in the graph. A non-negative search cost is given on each of the potential target vertices. The target is detected once the SDT searches its corresponding vertex. There may be some non-recoverable online blockages in the graph, but the existence of blockages is unknown to the SDT initially. If a blockage exists in the graph, it is disclosed online once the SDT visits one of its end-vertices. The graph stays connected when the blockages are omitted from it. The SDT begins from a certain vertex and aims to identify a route without any blocked edges which detects the target with minimum total traveling and search cost. We analyze this problem from a competitive analysis perspective under two scenarios with and without blockages. For the scenario with blockages, we provide a tight lower bound on the competitive ratio of deterministic solutions, an optimal deterministic solution, a randomized solution with a bounded expected competitive ratio, together with lower and upper bounds on the expected competitive ratio of the optimal randomized solutions. For the scenario without blockages, we provide tight lower bounds as well as optimal deterministic and randomized solutions.

Keywords Competitive analysis · Online blocked edges · Routing · Searching

✉ Davood Shiri
d.shiri@sheffield.ac.uk

Hakan Tozan
htozan@medipol.edu.tr

[1] Sheffield University Management School, University of Sheffield, Conduit Road, Sheffield S10 1FL, United Kingdom

[2] Department of Industrial Engineering, Istanbul Medipol University, Istanbul 34810, Turkey

1 Introduction

Routing problems under online uncertainty have been extensively investigated in recent years. In these problems, locations to be visited (Jaillet and Wagner 2008b, 2006) or the conditions of the roads (Akbari et al. 2021; Liu et al. 2021; Shiri et al. 2020) are unknown initially and are revealed incrementally over time. In fact, the assumption that the complete input information is available a priori is unrealistic in many real-world applications of routing problems (Jaillet and Wagner 2008a). For example, taxi and courier services necessitate an online model where locations to be visited are disclosed over time (Jaillet and Wagner 2006). As another example, in a post-disaster situation, the information about the damaged road segments is not available to the search-and-rescue team in the immediate emergency response phase and is disclosed in an online manner (Shiri and Salman 2020).

In this paper, with the aim of modeling routing problems to address real-world applications, we investigate an online optimization problem involving search defined on partially unknown road networks. In this problem, we study how to route a search-and-detection team (SDT) on a road network under edge uncertainty to search potential target locations in order to detect a static target, e.g., to detect and interdict a hostile object in a military operation. We model this problem on an undirected connected bounded graph $G = (V, E)$ which is given to the SDT as input. We note that modeling online routing problems on undirected graphs is a standard assumption in the literature, e.g., (Akbari and Shiri 2022; Liu et al. 2021; Zhang et al. 2019).

The graph G is partially unknown in the way that there exists a set of k ($k \geq 0$ is a bounded integer) non-recoverable online blocked edges $B \subset E$ which is unknown to the SDT initially. A blocked edge in B is disclosed online whenever one of its end-vertices is visited by the SDT. The graph G stays connected when all the potential blockages are omitted from it. Each edge $e \in E$ represents a transportation link and has a non-negative traveling cost d_e which is also certain to the SDT. Vertices in V are the collection of a source vertex $v_s \in V$ in which the SDT is initially positioned, set of potential target locations $V^t = \{v_1, v_2, \ldots, v_\lambda\} \subset V$ which is certain to the SDT, and the set of intersection points $I = V \setminus \{V^t \cup \{v_s\}\}$ which is also certain to the SDT. The set of intersection vertices represents the junction points in the road network, for example, an intersection vertex can be an end-vertex of a blockage. As another unknown factor in problem inputs, a static target is positioned at a vertex $v_* \in V^t$ which is unknown to the SDT initially. Each vertex $v \in V^t$ has a non-negative search cost s_v which is certain to the SDT and the target vertex v_* is disclosed online once the SDT visits v_* and incurs its search cost. The mission of the SDT is to identify a route without any blocked edges from v_s to v_* with minimum total traveling and search cost, i.e., the route ends as soon as the target is detected.

From this point onward, we refer to this problem as the *Online Routing and Searching Problem with k blocked edges* (k-ORSP). We present the list of used notation in the problem definition of the k-ORSP in Table 1. We do not assume that the number of blockages is certain to the SDT a priori. That is, we consider the number of blockages $k \geq 0$ as incomplete information in problem inputs. If a blockage exists in the graph, it is disclosed online once the SDT visits one of its end-vertices. Since the existence of the blockages is unknown to the SDT, we also investigate a special case of the problem

Table 1 Table of notation for the problem definition of the k-ORSP

Notation	Description
$G = (V, E)$	undirected graph, vertex set V, edge set E
$v_s \in V$	source vertex v_s
$V^t \subset V$ such that $v_s \notin V^t$, $\|V^t\| = \lambda$ and $\lambda \geq 1$	potential target vertices
$v_* \in V^t$	target vertex **(online uncertainty)**
$I = V \setminus \{V^t \cup \{v_s\}\}$	set of intermediate vertices
s_v for $v \in V^t$	search cost of vertex v
$B \subset E$	set of non-recoverable blockages **(online uncertainty)**
$k = \|B\|$ such that $(k \geq 0)$	number of blockages in the graph **(online uncertainty)**
d_e for $e \in E$	traveling cost of edge e

where there is no blockage in the graph, i.e., $k = 0$. We denote this special case of the k-ORSP by the ORSP henceforth.

We point out that in the k-ORSP, we consider two different types of online uncertainty, i.e., the non-recoverable blockages as well as the unknown target vertex. The k-ORSP has various real-world applications such as search-and-rescue as well as military operations. In the context of military operations, an online optimization model can be applied since the information about the blockages caused by an adversary (or enemy) and the target vertex (e.g., location of a hostile object or lost weapons) are not available to the security team in advance. In emergency search-and-rescue operations, the blocked links in the road network caused by a disaster as well as the location of the victims have an online nature and are unknown to the search-and-rescue team beforehand.

We survey the k-ORSP from a *competitive analysis* viewpoint applying the *competitive ratio* measure (Sleator and Tarjan 1985). That is, we compare the performance of online solutions which operate under incomplete information with the performance of the optimal solution provided in presence of complete input information from a worst-case perspective. We analyze both deterministic and randomized solutions using the competitive ratio criteria. For an online problem with a minimization objective function, the competitive ratio of a deterministic online solution is the maximum ratio of the cost of that online solution to the cost of the offline optimal solution over all problem instances. Similarly, the expected competitive ratio of a randomized online solution for an online minimization problem equals the maximum ratio of the expected cost of the online solution to the cost of the offline optimal solution over all problem instances. In the offline version of the k-ORSP, complete input information is available. Therefore, solving the offline version of this problem corresponds to finding the shortest path between v_s and v_* and hence the cost of the offline k-ORSP equals the cost of the cheapest path between v_s and v_* plus the search cost of v_*.

We conduct a competitive analysis on the k-ORSP according to the following two motivations. First, in many real-world applications of this problem such as military search-and-interdiction or emergency post-disaster search-and-rescue operations, it is difficult to capture probabilistic knowledge about the incomplete information. Second,

in such applications having a solution with a good worst-case performance is vital to tackle the malicious uncertainty. The organization of the rest of the article is as follows. In Sec. 1.1, we review the related articles to our study. In Sec. 1.2, we discuss the contributions of our paper. In Sec. 2, we give our results for deterministic solutions. In Sec. 3, we present our results for randomized solutions. We summarize our results in Sec. 4. We conclude our study and specify future research directions in Sec. 5.

1.1 Previous Studies

In this paper, we analyze the k-ORSP which comprises both edge and vertex uncertainties (i.e., the online blockages and the unknown target vertex) in the framework of the competitive analysis. Accordingly, the main focus of the literature review is on related works that apply competitive analysis methodology. We partition the literature review section into two parts. First, we review the relevant routing problems that comprise online blocked edges which are studied from a competitive analysis viewpoint. Next, we briefly present the related works which consider online searching problems on graphs based on the competitive ratio measure.

We note that there is a stream of online routing problems where requests for visits to vertices are revealed online to the traveler, e.g., (Jaillet and Wagner 2008a, b, 2006). In this stream of studies, the traveler should choose which requests to accept and provide an online solution to satisfy the admitted requests. These problems do not comprise online blockages in the graph or searching for an unknown target. Therefore, they are not directly related to the k-ORSP and we did not include them in our literature review.

1.1.1 Online routing problems with non-recoverable blockages

Routing problems comprising non-recoverable online blockages have a rich background in the literature. A closely related problem to the k-ORSP is the k-Canadian Traveler Problem (CTP). In fact, when the target vertex is known and its search cost is ignored in the k-ORSP, the problem reduces to the k-CTP. Papadimitriou and Yannakakis (1991) introduced the CTP which is a variant of the shortest path problem on graphs that contain online blockages and showed that providing an online solution that achieves a bounded competitive ratio is PSPACE-complete for the CTP. Westphal (2008) investigated the online k-CTP, a variant of the CTP with a fixed number of k online blocked edges. Westphal (2008) presented an optimal deterministic online solution, a tight lower bound of $2k+1$ on the competitive ratio of deterministic online solutions, and a lower bound of $k+1$ on the expected competitive ratio of randomized online solutions for the k-CTP. Xu et al. (2009) provided two deterministic online solutions for the k-CTP and compared them from competitive ratio viewpoint. Bender and Westphal (2015) introduced an optimal randomized solution for the online k-CTP for graphs having at most two online blockages and edge-disjoint paths. Shiri and Salman (2019b) presented an optimal randomized solution for the online k-CTP on graphs involving arbitrary number of online blockages and edge-disjoint paths. Recently, Demaine et al. (2021) proposed a randomized solution for the k-CTP which obtains an expected competitive ratio of $(1+\frac{\sqrt{2}}{2})k+\sqrt{2}$ in pseudo-polynomial time for arbitrary graphs.

Single-traveler routing problems with k non-recoverable online blocked edges have been studied with various objective functions from the competitive analysis perspective in the literature. If all or a subset of vertices must be visited in minimum total time, the k-ORSP resembles to variations of the Traveling Salemsman Problem (TSP) with k online blockages. Liao and Huang (2014) studied the Covering Canadian Traveler Problem (CCTP) which is a variant of the TSP on a complete edge-weighted graph and presented a deterministic touring solution within an $o(\sqrt{k})-$competitive ratio. Deterministic online solutions have been also analyzed from theoretical and computational competitive analysis perspectives for the Steiner TSP with k online blockages in which a subset of vertices must be visited (Zhang et al. 2016, 2015). Zhang and Xu (2018) studied deterministic solutions for the online covering salesman problem where the objective is to identify a tour such that each vertex in the graph is either on the tour or within a predetermined distance from a vertex on the tour.

If all or a subset of vertices must be visited such that the summation of the times of the visits are minimized, then the k-ORSP resembles to the variations of the Minimum Latency Problem (MLP) with k online blockages. Zhang et al. (2019) analyzed the online MLP (OMLP) with k online blocked edges where the objective is to minimize the summation of the times of the visits of the vertices. They investigated deterministic online solutions for this problem from theoretical and computational worst-case competitive ratio viewpoints. Akbari and Shiri (2021) studied a variant of the OMLP where priority weights are assigned to the vertices and the objective is to minimize the total weighted latency of vertices by investigating deterministic online solutions from worst-case competitive ratio point of view and proposing two efficient deterministic heuristic solutions tested on real city instances.

Here, we point out that our study differs from all the articles discussed above since we consider vertex uncertainty (i.e., unknown location of the target) in addition to online blocked edges in our problem inputs. Unlike the online versions of the TSP and the MLP with k online blockages where all or a subset of vertices should be visited, a solution of the k-ORSP terminates as soon as the target is detected at one of the potential target vertices.

In another related problem, Shiri et al. (2020) studied a search-and-rescue problem with multiple teams and non-recoverable online blockages in which a subset of vertices (called the critical vertices) must be serviced. They, considered online search-and-rescue times on the critical vertices in the sense that the required search-and-rescue time at a critical vertex is disclosed once it is visited by a search-and-rescue team. They offered two lower bounds on the competitive ratio of deterministic solutions and two heuristics that are computationally tested on random instances. Among the articles which are mentioned in this section, the work of Shiri et al. (2020) is the most related problem to our study since it considers both online blockages as well as uncertainty at the vertices. However, contrary to our study where a solution terminates once the target is detected, all the critical vertices must be visited and serviced in Shiri et al. (2020). Moreover, Shiri et al. (2020) focused on computational aspects of their studied problem and did not prove the competitive ratio of their suggested heuristics, whereas in here we provide a comprehensive competitive analysis for the k-ORSP by proposing tight lower bounds, an optimal deterministic solution with a proven competitive ratio, and a randomized solution with a bounded expected competitive ratio.

1.1.2 Online searching problems

Online searching problems in which a static target is positioned at a point of a graph are studied with respect to two different search paradigms in the literature. (1) *Pathwise search* where the searcher must identify a continuous walk to detect the target (Fleischer et al. 2009; Jaillet and Stafford 2001; Koutsoupias et al. 1996), and (2) *expanding search* where the searcher may restart from any previously searched point at any time (Angelopoulos and Lidbetter 2020; Angelopoulos et al. 2019). The k-ORSP corresponds to the pathwise search paradigm since it involves traveling costs on road networks. Hence, we confine our literature review on pathwise search problems that are investigated based on the competitive ratio measure.

The competitive ratio of online solutions in the pathwise search paradigm was first considered by Beck and Newman (1970) where the input graph is a real line. This problem was later studied on graphs having a star topology by Gal (1972, 1974). There are also several other works which consider deterministic and randomized solutions in the pathwise search context on a real line or on a star graph, e.g., (Demaine et al. 2006; Jaillet and Stafford 2001). In our study, we consider the k-ORSP on undirected graphs with an arbitrary topology where the input graph is not completely known in the sense that it may contain some non-recoverable blockages.

Koutsoupias et al. (1996) proved that it is NP-hard to approximate competitive ratio of deterministic solutions on fixed undirected graphs with unit length edges where the target is positioned at one of the vertices. In addition, they proposed constant-factor approximation deterministic and randomized solutions. Ausiello et al. (2000) extended the problem in (Koutsoupias et al. 1996) to edge-weighted graphs and demonstrated connections between graph searching and routing problems such as the TSP and the MLP. Fleischer et al. (2009) was the first study that considered an online searching problem on an unknown graph in which information about adjacent edges and vertices is disclosed once the searcher visits a new vertex. They proposed constant-factor approximation online solutions on undirected graphs where the target is permitted to position at any point along the edges of the graph. However, they discussed that no constant-factor approximation online solution exists for the problem where the potential target locations are restricted to the vertices. Angelopoulos and Lidbetter (2020) provided a constant-factor approximation randomized solution for pathwise search on a known graph. In their study, search costs are not incorporated with the potential target locations and the target is detected once the searcher arrives at its location.

Our study differs from all the aforementioned articles since we consider online blockages in the input graph. Among these articles, the work of Fleischer et al. (2009) is the most related work to the k-ORSP. Yet their study is significantly different from our work in terms of problem setting and results. Fleischer et al. (2009) considered a general problem setting in which the input graph is completely unknown, but we consider a partially unknown road network that contains online blockages. Moreover, we consider both edge traveling costs as well as search costs at potential target locations while Fleischer et al. (2009) only considered traveling costs in their study. Furthermore, we provide optimal online solutions for the k-ORSP in which the target location is restricted to vertices of the input graph, whereas Fleischer et al. (2009) showed that no

constant-factor approximation online solution exists when the potential target locations are restricted to the vertices in their problem setting.

1.2 Our Contributions

In order to complement the prior works discussed in the literature review section, we introduce the k-ORSP which comprises both vertex (unknown target vertex) and edge (online blockages) uncertainties to address real-world applications of routing and searching problems such as military interdiction as well as post-disaster search-and-rescue operations. Moreover, we provide a comprehensive competitive analysis for the k-ORSP and a reduced version of this problem in which there exists no blockage in the graph (i.e., the ORSP). For both of these problems, we specify tight lower bounds on the competitive ratio of deterministic online solutions (Lemma 2.1 and Corollary 2.1) with an optimal deterministic solution (Algorithm 1 and Theorem 2.1). For the k-ORSP, we derive lower and upper bounds on the expected competitive ratio of the optimal randomized solutions (Corollary 3.1 and Lemma 3.5) and introduce a randomized solution (Algorithm 2) with a bounded competitive ratio (Theorem 3.1). Using these results, we specify an interval for the competitive ratio of the optimal randomized solutions for the k-ORSP (Corollary 3.2). Furthermore, we introduce a tight lower bound on the expected competitive ratio of randomized online solutions (Lemma 3.1) with an optimal matching randomized solution for the ORSP (Algorithm 2) whose expected competitive ratio meets the given lower bound (Theorem 3.2). A summary of our results is presented in Table 2 which is described in detail in Sec. 4.

2 Deterministic online solutions

We first derive lower bounds on the competitive ratio of deterministic online solutions for the k-ORSP and the ORSP. Next, we present a deterministic online solution which meets our provided lower bounds for both the k-ORSP and the ORSP. In this way, we prove the tightness of the lower bounds as well as optimality of our deterministic online solution for both problems.

2.1 Lower bounds

We begin with proving the lower bound for the k-ORSP. We then conclude the lower bound for the ORSP from our proposed lower bound for the k-ORSP.

Table 2 Summary of results

Solution type	ORSP				k-ORSP			
	LB	UB	BS	CR	LB	UB	BS	CR
Deterministic	$2\lambda-1$	$2\lambda-1$	LCS	$2\lambda-1$	$2(k+\lambda)-1$	$2(k+\lambda)-1$	LCS	$2(k+\lambda)-1$
Randomized	λ	λ	SCS	λ	$max\{\lambda, k+1\}$	$2(k+\lambda)-1$	LCS	$2(k+\lambda)-1$

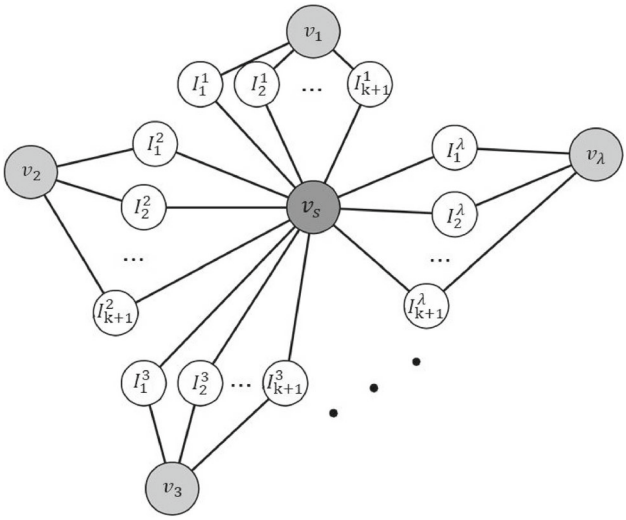

Fig. 1 The analyzed instance for the proof of the lower bound of $2(k+\lambda)$-1

Lemma 2.1 *There is no deterministic online solution whose competitive ratio is less than* $2(k+\lambda)-1$ *for the k-ORSP, where k and* λ *are the numbers of blocked edges and potential target vertices, respectively.*

Proof For any deterministic online solution (ALG), we show that there is an instance of inputs of the k-ORSP, such that the competitive ratio of ALG is not less than $2(k+\lambda)-1$ on the considered instance. To prove the lower bound, we consider the input graph shown in Fig. 1. In this graph v_s is the source vertex where the SDT is initially positioned, $V^t = \{v_1, v_2, \ldots, v_\lambda\}$ is the set of potential target vertices where $v_* \in \{v_1, v_2, \ldots, v_\lambda\}$, and $I = \{I_i^j \mid i = 1, 2, \ldots, k+1 \; and \; j = 1, 2, \ldots, \lambda\}$ is the set of intersection vertices. The traveling cost of edges (v_s, I_i^j) for $I_i^j \in I$ is assumed to be 1 and the traveling cost of edges (I_i^j, v_j) for $I_i^j \in I$ and $v_j \in \{v_1, v_2, \ldots, v_\lambda\}$ is set to ϵ. In addition, the search cost of all the vertices in $\{v_1, v_2, \ldots, v_\lambda\}$ is fixed to ϵ.

Any deterministic online solution executed on the instance shown in Fig. 1 can be considered as an iterative procedure such that at each iteration the SDT tries one of the paths $v_s - I_i^j - v_q$ ($i \in \{1, 2, \ldots, k+1\}$ and $j \in \{1, 2, \ldots, \lambda\}$) to arrive at v_q and search it. For an arbitrary deterministic solution, we consider the instance where the SDT finds the edge (I_i^j, v_j) blocked at the first k iterations, i.e., the SDT does not search any of the potential target vertices in V^t and has to go back to v_s by incurring a traveling cost of $2*1 = 2$ at the first k iterations. Moreover, for an arbitrary deterministic solution, we consider the instance in which the target vertex v_* is not one of the vertices which are searched at iterations $k+1$ to $k+\lambda-1$, i.e., the SDT has to go back to v_s by incurring a total traveling and search cost of $2(1+\epsilon)+\epsilon = 2+3\epsilon$ in these $\lambda - 1$ iterations.

Thus, in any deterministic online solution executed on the instance described above, the SDT detects the target vertex v_* at the $(k+\lambda)^{th}$ iteration where the total traveling

and search cost of the $(k+\lambda)^{th}$ iteration is $1+\epsilon+\epsilon = 1+2\epsilon$. Hence, the total traveling and search cost of an arbitrary deterministic solution on the described instance is at least $2(k) + (\lambda - 1)(2 + 3\epsilon) + 1 + 2\epsilon = 2(k + \lambda) + (3\lambda - 1)\epsilon - 1$. We point out in the offline optimal solution the target vertex v_* is certain to the SDT initially. Thus, the SDT takes the path with cost $1 + \epsilon$ from v_s to v_* and incurs a search cost of ϵ at v_*, i.e., the total cost of the offline optimal solution is $1 + 2\epsilon$. The proof is complete when ϵ approaches 0. □

Next, we apply the lower bound which we presented in Lemma 2.1 for the k-ORSP to derive the lower bound for the ORSP. We note that the ORSP is a reduced version of the k-ORSP when there is no blockage in the graph (i.e., $k = 0$).

Corollary 2.1 *There is no deterministic online solution whose competitive ratio is less than* $2\lambda - 1$ *for the ORSP.*

Proof The lower bound follows if we replace k with 0 in the proof of Lemma 2.1.

2.2 Optimal deterministic solution for the *k*-ORSP and the ORSP

We present a deterministic online solution which is optimal for both the k-ORSP and the ORSP. To introduce this solution, we need to provide the following definition.

Definition 2.1 We define the *operational cost* of v_j ($v_j \in V^t$) as the summation of the cost of the cheapest path between v_s and v_j plus the search cost of v_j. We represent the value of the operational cost of v_j by o_j for $v_j \in V^t = \{v_1, v_2, \ldots, v_\lambda\}$.

We refer to our deterministic solution as the *Lowest-Cost-Search* (LCS) algorithm. We note that the LCS is applicable to both the k-ORSP and the ORSP. The LCS is an iterative procedure in which the graph is updated at the end of each iteration. At the beginning of each iteration, the SDT computes the operational costs of the unsought potential target vertices in V^t on the updated graph and sorts them in non-decreasing order of their operational costs (Definition 2.1) in a list. Then, the SDT takes the cheapest path between v_s and the vertex with the least operational cost (i.e., the first element in the sorted list) in order to visit and search that vertex. If a blockage is found on the path, the SDT discards the found blockage from the graph, returns to v_s, and a new iteration begins. Otherwise, the SDT arrives at the potential unsought target vertex with the least operational cost and searches it. Then, if the target vertex v_* is detected, the LCS ends. Otherwise (if the target vertex v_* is not detected), the SDT removes the searched vertex from V^t, returns to v_s, and a new iteration begins. This procedure is repeated until the target vertex (v_*) is detected. The pseudocode of the LCS is provided in Algorithm 1.

The LCS terminates in at most $k + \lambda \leq |E| + |V|$ iterations. All pairs cheapest paths can be computed in $O(|V|^3)$ at each iteration (Chan 2010). Thus, the worst-case time-complexity of the LCS is at most $O(|E||V|^3 + |V|^4)$. We point out that the LCS can be used to solve both the k-ORSP and the ORSP, i.e., when the LCS is applied for the ORSP steps 8 to 13 are not implemented. Next, we show that the LCS is optimal for these problems.

Algorithm 1 : LCS Algorithm

Input:

a: $G = (V, E)$ ▷ initial input graph
b: $v_s \in V$ ▷ source vertex of the SDT
c: $V^t = \{v_1, v_2, \ldots, v_\lambda\}$ ▷ set of potential target vertices
d: s_v ▷ search cost of vertex $v \in V^t$
e: d_e ▷ traveling cost of edge $e \in E$

1: **Initiation:**
a: $F = \emptyset$ ▷ F : set of found blocked edges.
b: $q = 1$ ▷ q : iteration counter variable
c: $G_1 = G = (V, E)$ ▷ G_q : updated graph with revealed information at iteration q
d: $V^{'} = V^t$ ▷ list of unsought potential target vertices
e: flag = 1 ▷ flag: variable to terminate loops.
2: **While** flag **do:**
3: Calculate o_i^q for $v_i \in V^{'}$
▷ calculate the operational costs (Definition 2.1) of vertices in $V^{'}$ on the updated graph G_q
4: $L^q \leftarrow G_q$ ▷ compute sorted list of vertices in $V^{'}$ in non-decreasing order of operational costs on G_q
5: $v = L_1^q$ ▷ first element in L^q with the least operational cost
6: $SP^q(v_s, v)$ ▷ cheapest path from v_s to v on G_q
7: Take $SP^q(v_s, v)$ ▷ the SDT takes $SP^q(v_s, v)$
8: **if** $\exists\, e_q$ which is blocked on $SP^q(v_s, v)$ **then:** ▷ there is a blockage on $SP^q(v_s, v)$
9: $F = F \cup e_q$
10: $q = q + 1$
11: $G_q = (V, E \setminus F)$
12: return to source vertex v_s.
13: Go to step 3.
14: **Else:**
15: take $SP^q(v_s, v)$ to reach v and search v.
16: **if** $v = v_*$ **then:** ▷ the target vertex v_* is detected by the SDT
17: flag = 0 ▷ Termination Point.
18: **Else;**
19: $V^{'} = V^{'} \setminus \{v\}$
20: $q = q + 1$
21: $G_q = G_{q-1}$
22: return to source vertex v_s.
23: Go to step 3.
24: **end if**
25: **end if**
26: **end while**

Theorem 2.1 *The competitive ratio of the LCS is* $2(k + \lambda) - 1$ *and* $2\lambda - 1$ *for the k-ORSP and the ORSP, respectively.*

Proof We refer to the steps of Algorithm 1 in our proof. At iteration q ($q \geq 1$) of the LCS, the SDT sorts the unsought potential target vertices in non-decreasing order of their operational costs and chooses a vertex $v = L_1^q$ with the least operational cost (step 5). We denote the operational cost of vertex v which is chosen at iteration q by o^q in our proof. Next, the SDT takes the cheapest path between v_s and v (steps 3 to 7). Then, three scenarios may occur.

- Scenario 1. The SDT finds a blockage on the graph before arriving at v and returns to v_s without searching v (steps 8 to 12). In this case, the iteration cost is at most $2o^q$.
- Scenario 2. The SDT arrives at v and searches the vertex. In this case, the target is not detected and the SDT returns to v_s (steps 18 to 22). Thus, the iteration cost is at most $2o^q$.
- Scenario 3. The SDT arrives at v and detects the target after searching v (steps 15 to 17). In this case, the LCS terminates and the iteration cost is o^q.

Suppose that the target is detected at iteration q_*. The cost of the offline optimal solution equals o^{q_*} since the LCS searches the vertices in non-decreasing order of their operational costs at each iteration. It holds that o^q is non-decreasing in q ($q \leq q_*$) because at each iteration one blockage is eliminated from the graph or one vertex is removed from the set of potential target vertices, i.e.,

$$o^1 \leq o^2 \leq \ldots \leq o^{q_*}.$$

Hence, the total cost of the LCS is at most

$$\sum_{q=1}^{q_*-1} 2o^{q_*} + o^{q_*}.$$

We note that the LCS ends in at most $k + \lambda$ iterations applied to an instance of the k-ORSP, i.e., $q_* \leq k + \lambda$ for the k-ORSP. Thus, the total cost of the LCS is at most $2((k + \lambda) - 1)o^{q_*}$. The theorem follows for the k-ORSP, since the cost of the offline optimal solution is o^{q_*}. Also, the theorem follows for the ORSP when the value of k is set to 0 in the proof.

3 Randomized online solutions

We first prove lower bounds on the expected competitive ratio of randomized online solutions for the k-ORSP and the ORSP. Then, we introduce a randomized solution which is applicable for both problems. We show that our randomized solution achieves a bounded expected competitive ratio for the k-ORSP while it is optimal for the ORSP. Finally, we propose upper bounds on the expected competitive ratio of the optimal randomized solutions for the k-ORSP.

3.1 Lower bounds

Here, we first derive two different lower bounds for the expected competitive ratio of randomized solutions for the k-ORSP by considering different instances of this problem. By combining these two lower bounds, we prove a tighter lower bound for the expected competitive ratio of randomized solutions for the k-ORSP. We note that, since the ORSP is a reduced version of the k-ORSP, our lower bounds also hold for the ORSP when we set $k = 0$ in our proofs.

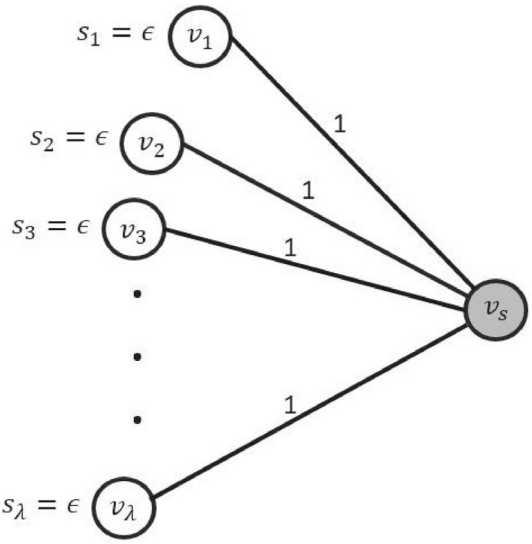

Fig. 2 The analyzed instance for the proof of Lemma 3.1

We begin with analyzing an instance of the k-ORSP in the absence of blockages, i.e., an ORSP instance. Therefore, the lower bound holds for both the k-ORSP and the ORSP.

Lemma 3.1 *There is no randomized online solution whose expected competitive ratio is less than λ for the k-ORSP and the ORSP.*

Proof We utilize Yao's principle (Yao et al. 1977) to derive the lower bound. For that, we consider the instance shown in Fig. 2. In this graph, v_s is the source vertex of the SDT, $V' = \{v_1, v_2, \ldots, v_\lambda\}$ are the potential target vertices, and the set of intersection vertices I is empty. The traveling costs of the edges (v_s, v_j) is 1 for $j = 1, 2, \ldots, \lambda$. Also, the search costs of the vertices $v_1, v_2, \ldots, v_\lambda$ is set to ϵ. We assume that there is no blockage in the graph, i.e., $k = 0$. We enforce the target vertex v_* be one of the vertices in $\{v_1, v_2, \ldots, v_\lambda\}$ according to uniform probability distribution.

We note that any deterministic solution executed on the instance described above can be considered as an iterative procedure such that at each iteration the SDT tries one of the edges (v_s, v_j) $(j \in \{1, 2, \ldots, \lambda\})$ to search one of the unsought vertices in $\{v_1, v_2, \ldots, v_\lambda\}$. If the target vertex is detected, the solution terminates and the iteration cost would be $1 + \epsilon$. Otherwise, the SDT returns to v_s and the iteration cost would be $2 + \epsilon$.

We note that in any deterministic solution, the SDT detects the target at iteration $j \in \{1, 2, \ldots, \lambda\}$ with probability $\frac{1}{\lambda}$ because we enforced uniform probability distribution on the location of the target. If the procedure terminates at iteration $j \in \{1, 2, \ldots, \lambda\}$, the SDT incurs a total cost of $2j + j\epsilon - 1$. Thus, the expected cost would be

$$\frac{1}{\lambda}\sum_{j=1}^{\lambda}(2j + j\epsilon - 1) = \frac{1}{\lambda}\sum_{j=1}^{\lambda}(2j - 1) + \frac{1}{\lambda}\sum_{j=1}^{\lambda} j\epsilon = \lambda + \frac{\lambda + 1}{2}\epsilon.$$

We proved that the expected cost of any deterministic solution against the uniform probability distribution enforced on the location of the target is $\lambda + \frac{\lambda+1}{2}\epsilon$. We point out that the cost of the offline optimal solution is $1 + \epsilon$. The instance shown in Fig. 2 represents an instance of both the k-ORSP and the ORSP. The lemma follows for both problems by Yao's principle (Yao et al. 1977) when ϵ approaches 0.

Next, we consider an instance of the k-ORSP with only one potential target vertex (i.e., $|V^t| = 1$) whose search cost is a sufficiently small positive value. As discussed in the literature review section, when the target vertex is known ($|V^t| = 1$) and its search cost is ignored, the k-ORSP reduces to the k-CTP.

Lemma 3.2 *There is no randomized online solution whose expected competitive ratio is less than $k + 1$ for the k-ORSP.*

Proof The k-CTP is a reduced version of the k-ORSP in which the target vertex is certain a priori and the search cost of the target vertex is 0. The lemma holds, because no randomized online solution obtains an expected competitive ratio better than $k + 1$ for the k-CTP according to Westphal (2008).

Intuitively, the lower bound of $k + 1 = 0 + 1 = 1$ also holds for the ORSP since $k = 0$ in this problem. By combining the results in Lemmas 3.1 and 3.2, we prove a tighter lower bound for the k-ORSP.

Corollary 3.1 *There is no randomized online solution whose expected competitive ratio is less than*

$$max\{\lambda, k + 1\}$$

for the k-ORSP.

3.2 Randomized solution for the *k*-ORSP and the ORSP

We first introduce a randomized solution that can be used for both the k-ORSP and the ORSP which is based on a specific probability distribution (Lemma 3.3). Next, we prove a bounded expected competitive ratio for our randomized solution for the k-ORSP. Furthermore, we show that our randomized solution is optimal for the ORSP. For presenting our results, we define the *similar costs feature* for the vertices in the set of potential target vertices $V^t = \{v_1, v_2, \ldots, v_\lambda\}$ as follows.

Definition 3.1 Vertices $v_1, v_2, \ldots, v_\lambda$ with operational costs $o_1 \leq o_2 \leq \ldots \leq o_\lambda$ (Definition 2.1), have the similar costs feature if

$$o_i \leq \frac{2}{\lambda} \sum_{j=1}^{\lambda} o_j,$$

for $i = 1, 2, \ldots, \lambda$.

For a set of numbers that have the similar costs feature, Bender and Westphal (2015) proved that the probability distribution presented in the following lemma exists.

Lemma 3.3 *For numbers $o_1 \leq o_2 \leq \ldots \leq o_\lambda$ with similar costs feature, the probability distribution $\Omega_\lambda = \Lambda^* p^{'}$ in which Ω_λ and $p^{'}$ are λ-vectors, $\Lambda^* = \sum_{i=1}^{\lambda} \frac{1}{p_i^{'}} \in [0, 1]$,*

$$p_i^{'} = \frac{(2-\lambda)o_i + \sum_{j=1\, j\neq i}^{n} 2o_j}{\lambda^2 o_i} \quad \forall i = 1, 2, \ldots, \lambda$$

exists and belongs to the polyhedron

$$Q_\lambda = \{p \in \mathbb{R}_+^n : (2-\lambda)p_i + \sum_{j=1\, j\neq i}^{\lambda} 2\frac{o_j}{o_i} p_j \leq 1 \ \ \forall i = 1, 2, \ldots, \lambda, \sum_{i=1}^{\lambda} p_i = 1\}.$$

We introduce a randomized solution which is applicable for both the k-ORSP and the ORSP that utilizes the probability distribution in Lemma 3.3 to assign probabilities to vertices that have the similar costs feature. We refer to our randomized solution as the *Similar-Costs-Search* (SCS) algorithm. The SCS terminates in at most $k + \lambda$ iterations. At each iteration, the SDT makes a list of unsought vertices that have the similar costs feature. Then, picks out of one of the vertices in the list with respect to the probability distribution given in Lemma 3.3, and takes the cheapest path between v_s and the chosen vertex. If a blockage is found, the SDT discards the found blockage from the graph, returns to v_s, and a new iteration begins. Otherwise, the SDT arrives at the chosen potential unsought target vertex and searches it. Then, if the target vertex v_* is detected, the SCS ends. Otherwise, the SDT returns to v_s, updates the list of unsought vertices, and a new iteration begins. This randomized process is repeated until the target vertex (v_*) is detected. The pseudocode of the SCS is given in Algorithm 2. The SCS terminates in at most $k + \lambda \leq |E| + |V|$ iterations. To compute the list of unsought vertices with similar costs feature with respect to their operational costs, all pairs cheapest paths can be computed in $O(|V|^3)$ at each iteration (Chan 2010), and sorting can be performed in $O(|V|log|V|)$, i.e., the list can be computed in $O(|V|log|V| + |V|^3) \simeq O(|V|^3)$ at each iteration. Thus, the worst-case time-complexity of the SCS is at most $O(|E||V|^3 + |V|^4)$.

To derive the competitive ratio of the SCS for the k-ORSP and the ORSP, we apply the following law which is proven in Shiri and Salman (2019a) about the numbers with similar costs feature.

Lemma 3.4 *If a number $o_{\lambda+1}$ has not the similar costs feature with numbers $o_1, o_2, \ldots, o_\lambda$ which have the similar costs feature, then $2\sum_{i=1}^{\lambda} p_i o_i < o_{\lambda+1}$.*

In the next theorem, we show that the SCS obtains a bounded expected competitive ratio for the k-ORSP.

Theorem 3.1 *The SCS has an expected competitive ratio of at most $(k+\lambda-1)(1+\lambda)+1$ for the k-ORSP.*

Algorithm 2 : SCS Algorithm

Input:

a: $G = (V, E)$ ▷ initial input graph
b: $v_s \in V$ ▷ source vertex of the SDT
c: $V^t = \{v_1, v_2, \ldots, v_\lambda\}$ ▷ set of potential target vertices
d: s_v ▷ search cost of vertex $v \in V^t$
e: d_e ▷ traveling cost of edge $e \in E$

1: **Initiation:**
a: $F = \emptyset$ ▷ F : set of found blocked edges.
b: $q = 1$ ▷ q : iteration counter variable
c: $G_1 = G = (V, E)$ ▷ G_q : updated graph with revealed information at iteration q
d: $V^{'} = V^t$ ▷ list of unsought potential target vertices
e: flag = 1 ▷ flag: variable to terminate loops.
2: **While** flag **do:**
3: Calculate o_i^q for $v_i \in V^{'}$
▷ calculate the operational costs (Definition 2.1) of vertices in $V^{'}$ on G_q
4: $L_{SCF}^q \leftarrow V^{'}$ ▷ compute the list of vertices in $V^{'}$ that have the similar costs feature
5: $\eta = |L_{SCF}^q|$ ▷ denote the number of added vertices to L_{SCF}^q by η
6: $v \leftarrow \Omega_\eta \leftarrow L_{SCF}^q$
▷ pick out a vertex from L_{SCF}^q with respect to Ω_η (probability distribution in Lemma 3.3)
7: $SP^q(v_s, v)$ ▷ cheapest path from v_s to v on G_q
8: Take $SP^q(v_s, v)$ ▷ the SDT takes $SP^q(v_s, v)$
9: **if** $\exists\ e_q$ which is blocked on $SP^q(v_s, v)$ **then:** ▷ there is a blockage on $SP^q(v_s, v)$
10: $F = F \cup e_q$
11: $q = q + 1$
12: $G_q = (V, E \setminus F)$
13: return to source vertex v_s.
14: Go to step 3.
15: **Else:**
16: take $SP^q(v_s, v)$ to reach v and search v.
17: **if** $v = v_*$ **then:** ▷ the target vertex v_* is detected by the SDT
18: flag = 0 ▷ Termination Point.
19: **Else:**
20: $V^{'} = V^{'} \setminus \{v\}$
21: $q = q + 1$
22: $G_q = G_{q-1}$
23: return to source vertex v_s.
24: Go to step 3.
25: **end if**
26: **end if**
27: **end while**

Proof We refer to the steps of Algorithm 2 in our proof. We note that the SCS terminates in at most $k + \lambda$ iterations. Suppose that $v_* \in \{v_1, v_2, \ldots, v_\lambda\}$ is the target vertex. We represent the cost of the offline optimal solution by o_*. At the beginning of iteration $q \in \{1, 2, \ldots, k+\lambda\}$, the SDT makes a list L_{SCF}^q which contains the unsought vertices in V^t with similar costs feature (step 4). We denote the size of $L_{SCF}^q = \{v_1, v_2, \ldots, v_\eta\}$ by $\eta \leq \lambda$ (step 5). Then, the SDT picks out a vertex v from L_{SCF}^q with respect to the probability distribution $\Omega_\eta = (p_1, p_2, \ldots, p_\eta)$ (Lemma 3.3 and step 6), and takes the cheapest path from v_s to $v \in L_{SCF}^q$ (step 8). If a blockage is found on the path, the SDT updates the graph and returns to v_s (steps 9 to 14). Otherwise, the SDT arrives

at v and searches the vertex (step 16). If the target is detected, the SCS terminates (steps 17-18). Otherwise, the SDT updates the list of unsought vertices and returns to v_s (steps 20-23). If the target is not detected at iteration q, an expected competitive ratio of at most

$$2\sum_{i=1}^{\eta} p_i \frac{o_i^q}{o_*}$$

is incurred, i.e., o_i^q is the operational cost of vertex $v_i \in L_{SCF}^q$. We propose the proof by upper bounding the incurred expected competitive ratio at each iteration of the SCS procedure. We consider two scenarios.

- Scenario 1. v_* has the similar costs feature with the vertices in L_{SCF}^q. Because of the existence of the blockages in G_q, there exists some $v_{i^*} \in L_{SCF}^q = \{v_1, v_2, \ldots, v_\eta\}$ such that $o_{i^*}^q \leq o_*$ ($o_{i^*}^q \in \{o_1^q, o_2^q, \ldots, o_\eta^q\}$), i.e., if no blockage exists in G_q, then there exists $v_{i^*} \in L_{SCF}^q = \{v_1, v_2, \ldots, v_\eta\}$ such that $o_{i^*}^q = o_*$. To upper bound the expected competitive ratio at iteration q, we replace o_* by $o_{i^*}^q$ in the denominator to obtain

$$2\sum_{i=1}^{\eta} p_i \frac{o_i^q}{o_{i^*}^q} = 2p_{i^*} + 2\sum_{i=1, i\neq i^*}^{\eta} p_i \frac{o_i^q}{o_{i^*}^q}.$$

  Since $v_1, v_2, \ldots, v_\eta$ have the similar costs feature, it holds that

$$2p_{i^*} + 2\sum_{i=1, i\neq i^*}^{\eta} p_i \frac{o_i^q}{o_{i^*}^q} \leq 1 + \eta(p_{i^*}) \leq 1 + \eta,$$

  for all $i^* \in \{1, 2, \ldots, \eta\}$ according to Lemma 3.3. Also, we note that $\eta \leq \lambda$, hence the expected competitive ratio of iteration q would be at most $1 + \lambda$.
- Scenario 2. v_* does not have the similar costs feature with the vertices in L_{SCF}^q. Since v_* does not have the similar costs feature with the vertices in L_{SCF}^q, $2\sum_{i=1}^{\eta} p_i o_i < o_*$ according to Lemma 3.4. Thus, the incurred expected competitive ratio at iteration q would be at most 1 in this scenario.

To enforce a higher upper bound on the expected competitive ratio of the SCS, we assume that Scenario 1 occurs at the first $k + \lambda - 1$ iterations. Accordingly, the total expected competitive ratio of the SCS can be upper bounded by $(k + \lambda - 1)(1 + \lambda)$ in these $k + \lambda - 1$ iterations. Also, the incurred expected competitive ratio at the $(k + \lambda)^{th}$ iteration is $\frac{o_*}{o_*} = 1$ since the target is detected at this iteration. The theorem follows. □

Although the SCS obtains a bounded expected competitive ratio for the k-ORSP, its performance decreases when k increases. However, the SCS performs optimally in a reduced version of the k-ORSP in which there is no blockage in the graph. In the next theorem, we prove the optimality of the SCS for the ORSP.

Theorem 3.2 *The expected competitive ratio of the SCS is λ for the ORSP.*

Proof We refer to the steps of Algorithm 2 in our proof. Suppose that $V^t = \{v_1, v_2, \ldots, v_\lambda\}$ is the set of potential target vertices. We note that in the ORSP, there is no blockage in the graph, i.e., $k = 0$. Hence, steps 9 to 14 are not implemented when the SCS is executed on an instance of the ORSP. We apply induction on the number of potential target vertices λ.

- **Base case.** If $\lambda = 1$, the SDT picks out the only potential target vertex v_1 with probability $p_1 = 1$. Then, takes the cheapest path between v_s and v_1, searches the vertex, and detects the target. Thus, the expected cost of the SCS equals the operational cost of v_1, i.e., o_1 (Definition 2.1). Hence, the cost of the offline optimal solution is also o_1. Thus the expected competitive ratio would be 1.
- **Induction.** For $\lambda \geq 2$, suppose that $o_1 \leq o_2 \leq \ldots \leq o_\lambda$ represent the operational costs of the vertices in $V^t = \{v_1, v_2, \ldots, v_\lambda\}$. Also, suppose that v_{i^*} ($i^* \in \{1, 2, \ldots, \lambda\}$) is the target vertex. Therefore, the cost of the offline optimal solution is o_{i^*}. At iteration 1 of the SCS, the SDT makes a list L^1_{SCF} which contains the vertices in V^t with similar costs feature (step 4). We denote the size of L^1_{SCF} by η (step 5). We consider two scenarios.
  - Scenario 1. $v_{i^*} \in L^1_{SCF}$. The SDT picks out a vertex v from L^1_{SCF} based on the probability distribution $\Omega_\eta = (p_1, p_2, \ldots, p_\eta)$ (Lemma 3.3 and step 6), takes the cheapest path from v_s to $v \in L^1_{SCF}$ (step 8), and searches the vertex (step 16). If the target is detected, the SCS terminates and the expected competitive ratio would be 1 (steps 17-18). Otherwise, the SDT updates the list of unsought vertices and returns to v_s (steps 20-23). In this case, we represent the cost of the SCS from the time that the SDT arrives at v_s until the termination time of the SCS by $C^{\lambda-1}$. Because $\eta \leq \lambda$, the expected competitive ratio of the SDT is at most

$$p_{i^*} + \sum_{j=1\ j\neq i^*}^{\eta} p_j \frac{2o_j + C^{\lambda-1}}{o_{i^*}} \leq p_{i^*} + \sum_{j=1\ j\neq i^*}^{\lambda} p_j \frac{2o_j + C^{\lambda-1}}{o_{i^*}},$$

where $C^{\lambda-1} \leq (\lambda - 1)c_{i^*}$ by the induction assumption. Accordingly, the expected competitive ratio is not greater than

$$p_{i^*} + \sum_{j=1\ j\neq i^*}^{\lambda} p_j \left(\frac{2o_j}{o_{i^*}} + \lambda - 1\right) = p_{i^*} + \sum_{j=1\ j\neq i^*}^{\lambda} \left(p_j \frac{2o_j}{o_{i^*}}\right) + \left((\lambda-1) \sum_{j=1\ j\neq i^*}^{\lambda} p_j\right).$$

We must show that the expected competitive ratio is not greater than λ, i.e.,

$$p_{i^*} + \sum_{j=1\ j\neq i^*}^{\lambda} \left(p_j \frac{2o_j}{o_{i^*}}\right) + \left((\lambda - 1) \sum_{j=1\ j\neq i^*}^{\lambda} p_j\right) \leq \lambda,$$

or,

$$p_{i^*} + \sum_{j=1\ j\neq i^*}^{\lambda} \left(p_j \frac{2o_j}{o_{i^*}} \right) \leq \lambda - \left((\lambda - 1) \sum_{j=1\ j\neq i^*}^{\lambda} p_j \right).$$

We deduct $(\lambda - 1)p_{i^*}$ from both sides to obtain

$$(2 - \lambda)p_{i^*} + \sum_{j=1\ j\neq i}^{\lambda} p_j \frac{2o_j}{o_{i^*}} \leq 1,$$

which holds for all $i^* \in \{1, 2, \ldots, \lambda\}$ according to Lemma 3.3 because $\Omega_\lambda = (p_1, p_2, \ldots, p_\lambda) \in Q_\lambda$.

– Scenario 2. $v_{i^*} \notin L^1_{SCF}$. The SDT picks out a vertex v from L^1_{SCF} based on the probability distribution $\Omega_\eta = (p_1, p_2, \ldots, p_\eta)$ (Lemma 3.3 and step 6), takes the cheapest path from v_s to $v \in L^1_{SCF}$ (step 8), and searches the vertex (step 16). Since $v_{i^*} \notin L^1_{SCF}$, the target is not detected and the SDT returns to v_s by incurring a total cost of at most $2o_v$ (steps 20-23). Thus, an expected cost of less than or equal to $2\sum_{i=1}^{\eta} p_i o_i$ is incurred, i.e., o_i is the operational cost of vertex $v_i \in L^1_{SCF}$. We point out that $2\sum_{i=1}^{\eta} p_i o_i < o_{i^*}$ according to Lemma 3.4 because $v_{i^*} \notin L^1_{SCF}$, i.e., v_{i^*} has not similar costs feature with the vertices in L^1_{SCF}. Similar to Scenario 1, we represent the cost of the SCS from the time that the SDT arrives at v_s until the termination time of the SCS by $C^{\lambda-1}$ which is less than $(\lambda - 1)o_{i^*}$ by the induction assumption. It follows that the expected cost of the SCS is at most $o_{i^*} + (\lambda - 1)o_{i^*} = (\lambda)o_{i^*}$.

The theorem follows. □

3.3 Upper bound on the expected competitive ratio of optimal randomized solutions for the *k*-ORSP

Utilizing our results in previous sections, we present an upper bound on the expected competitive ratio of the optimal randomized solutions for the k-ORSP.

Lemma 3.5 *The expected competitive ratio of an optimal randomized solution for the k-ORSP is not greater than $min\{2(k+\lambda)-1, (k+\lambda-1)(1+\lambda)+1\} = 2(k+\lambda)-1$.*

Proof The SCS has an expected competitive ratio of at most $(k+\lambda-1)(1+\lambda)+1$ which suggests an upper bound on the expected competitive ratio of randomized solutions for the k-ORSP. We point out that any deterministic solution can be regarded as a randomized solution, designed only with probabilities 1 or 0. Thus, the competitive ratio of $2(k + \lambda) - 1$ for the LCS also suggests an upper bound on the expected competitive ratio of randomized solutions for the k-ORSP. We can conclude an upper bound of $min\{2(k + \lambda) - 1, (k + \lambda - 1)(1 + \lambda) + 1\} = 2(k + \lambda) - 1$ on the expected competitive ratio of the optimal randomized online solutions for the k-ORSP since $k \geq 0$ and $\lambda \geq 1$. □

Corollary 3.2 *An optimal randomized solution for the k-ORSP has an expected competitive ratio between* λ *and* $2(k + \lambda) - 1$.

4 Summary of competitive analysis results

A summary of competitive analysis results for the k-ORSP and the ORSP is given in Table 2. The results for the k-ORSP and the ORSP are differentiated in this table. Also, the results for deterministic and randomized solutions are categorized in different rows. In Table 2, columns "LB" and "UB" represent lower bound and upper bound on the competitive ratio of online solutions, respectively. The "BS" and "CR" columns denote the best found solution in our paper and its corresponding competitive ratio, respectively. As it can be observed, for the ORSP and deterministic solutions, the lower bound of $2\lambda-1$ is tight and the LCS is a matching optimal solution. Similarly, for the k-ORSP and deterministic solutions, the lower bound of $2(k+\lambda)-1$ is tight and the LCS is a matching optimal solution. For the ORSP and randomized solutions, the lower bound of λ is tight and the SCS is the optimal randomized solution that meets the lower bound of λ. For the k-ORSP and randomized solutions, $max\{\lambda, k+1\}$ and $2(k+\lambda)-1$ are the lower and upper bounds on the expected competitive ratio of the optimal randomized solutions, respectively, where the LCS is the solution whose competitive ratio matches the upper bound. An open research problem is to propose a randomized solution for the k-ORSP with an expected competitive ratio less that $2(k + \lambda) - 1$.

5 Conclusions and future research

We investigated two versions of an online routing and searching problem modelled on undirected graphs by considering two different types of online uncertainty defined on the vertices and the edges. In the first version (i.e., the k-ORSP), the target vertex is unknown and potential target vertices are associated with a search cost where there are some non-recoverable blockages in the graph which are unknown a priori such that each of them is disclosed once one of its end-vertices is visited. The objective in the k-ORSP is to identify a routing and searching policy which detects the target with minimum total traveling and search cost. For deterministic solutions for the k-ORSP, we proposed a tight lower bound and an optimal online solution. For randomized solutions for the k-ORSP, we specified lower and upper bounds on the expected competitive ratio of the optimal online solutions together with a randomized solution with a bounded expected competitive ratio. The second version (i.e., the ORSP) is a reduced variant of the k-ORSP where there is no blockage in the graph. For this problem, we proposed tight lower bounds and optimal solutions for both deterministic and randomized cases. These results are summarized in Table 2.

A future research direction is to analyze variations of the k-ORSP involving multiple teams and different levels of communication between the teams from competitive analysis viewpoint. An open research question is to specify a tight lower bound on the

expected competitive ratio of randomized solutions as well as an optimal randomized solution for the k-ORSP.

References

Akbari Vahid, Shiri Davood (2021) Weighted online minimum latency problem with edge uncertainty. *Eur. J. Oper. Res.*

Akbari Vahid, Shiri Davood (2022) An online optimization approach for post-disaster relief distribution with online blocked edges. Comput. & Oper. Res. 137:105533

Akbari Vahid, Shiri Davood, Sibel Salman F (2021) An online optimization approach to post-disaster road restoration. Trans. Res. Part B: Meth. 150:1–25

Angelopoulos Spyros, Lidbetter Thomas (2020) Competitive search in a network. Eur. J. Oper. Res. 286:781–790

Angelopoulos Spyros, Dürr Christoph, Lidbetter Thomas (2019) The expanding search ratio of a graph. Dis. Appl. Math. 260:51–65

Ausiello Giorgio, Leonardi Stefano, Marchetti-Spaccamela Alberto (2000) On Salesmen, Repairmen, Spiders, and Other Traveling Agents. *Pages 1–16 of:* Bongiovanni, Giancarlo, Petreschi, Rossella, & Gambosi, Giorgio (eds), *Algorithms and Complexity*. Berlin, Heidelberg: Springer Berlin Heidelberg

Beck Anatole, Newman DJ (1970) Yet more on the linear search problem. Israel J. Math. 8:419–429

Bender Marko, Westphal Stephan (2015) An optimal randomized online algorithm for the k-Canadian Traveller Problem on node-disjoint paths. J. Comb. Optim. 30:87–96

Chan Timothy M (2010) More Algorithms for All-Pairs Shortest Paths in Weighted Graphs. SIAM J. Comput. 39(5):2075–2089

Demaine Erik D, Fekete Sandor P, Gal Shmuel (2006) Online searching with turn cost. Theor. Comput. Sci. 361:342–355

Demaine Erik D, Huang Yamming, Liao Chung-Shou, Sadakane Kunihiko (2021) Approximating the Canadian Traveller Problem with Online Randomization. Algorithmica 83:1524–1543

Fleischer Rudolf, Kamphans Tom, KLEIN, Rolf, Langetepe, Elmar, & Trippien, Gerhard. (2009) Competitive online approximation of the optimal search ratio. SIAM J. Comput. 38(3):881–898

Gal Shmuel (1972) A general search game. Israel J. Math. 12:32–45

Gal Shmuel (1974) Minimax Solutions for Linear Search Problems. SIAM J. Appl. Math. 27(1):17–30

Jaillet P, Wagner MR (2008a) Online Vehicle Routing Problems: A Survey. *Chap. 10, pages 221–237 of:* Golden B, Raghavan S, Wasil E (ed), The Vehicle Routing Problem: Latest Advances and New Challenges. Boston, MA: Springer

Jaillet Patrick, Stafford Matthew (2001) Online Searching. Oper. Res. 49:501–515

Jaillet Patrick, Wagner Michael R (2006) Online Routing Problems: Value of Advanced Information as Improved Competitive Ratios. Transp. Sci. 40(2):200–210

Jaillet Patrick, Wagner Michael R (2008) Generalized Online Routing: New Competitive Ratios, Resource Augmentation, and Asymptotic Analyses. Oper. Res. 56:745–757

Koutsoupias Elias, Papadimitriou Christos, Yannakakis Mihalis (1996) Searching a fixed graph. *Pages 280–289 of:* Meyer, Friedhelm, & Monien, Burkhard (eds), Automata, Languages and Programming. Berlin, Heidelberg: Springer Berlin Heidelberg

Liao Chung-Shou, Huang Yamming (2014) The Covering Canadian Traveler Problem. Theor. Comput. Sci. 530:80–88

Liu Henan, Zhang Huili, Xu Yi (2021) The m-Steiner Traveling Salesman Problem with online edge blockages. J. Comb. Optim

Papadimitriou Christos, Yannakakis Mihalis (1991) Shortest paths without a map. Theor. Comput. Sci. 84:127–150

Shiri Davood, Salman F. Sibel (2019) Competitive analysis of randomized online strategies for the online multi-agent *k*-Canadian Traveler Problem. J. Comb. Optim. 37:848–865

Shiri Davood, Salman F. Sibel (2019) On the randomized online strategies for the *k*-Canadian traveler problem. J. Comb. Optim. 38:254–267

Shiri Davood, Salman F. Sibel (2020) Online Optimization of First-responder Routes in Disaster Response Logistics. IBM J. Res. Dev. 64:1–9

Shiri Davood, Akbari Vahid, Salman F. Sibel (2020) Online routing and scheduling of search-and-rescue teams. OR Spectrum 42(3):755–784

Sleator Daniel, Tarjan Robert (1985) Amortized efficiency of list update and paging rules. Com. ACM 28:202–208

Westphal Stephan (2008) A note on the k-Canadian Traveller Problem. Inf. Processing Letters 106(3):87–89

Xu Yinfeng, Hu Maolin, Su Bing, Zhu Binhai, Zhu Zhijun (2009) The Canadian Traveller Problem and its competitive analysis. J. Com. Optim. 18:195–205

Yao Andrew Chi-Chin. (1977) Probabilistic computations: Toward a unified measure of complexity. *Pages 222–227 of:* 18th Ann. Symp. Found. Comput. Sci. (sfcs 1977)

Zhang Huili, Xu Yinfeng (2018) Online covering salesman problem. J. Comb. Optim. 35:941–954

Zhang Huili, Tong Weitian, Xu Yinfeng, Lin Guohui (2015) The Steiner Traveling Salesman Problem with online edge blockages. Eur. J. Operat. Res. 243:30–40

Zhang Huili, Tong Weitian, Xu Yinfeng, Lin Guohui (2016) The Steiner Traveling Salesman Problem with online advanced edge blockages. Comput. & Operat. Res. 70:26–38

Zhang Huili, Tong Weitian, Lin Guohui, Xu Yinfeng (2019) Online minimum latency problem with edge uncertainty. Eur. J. Operat. Res. 273:418–429