Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Approximation and complexity of multi-target graph search and the Canadian traveler problem

Martijn van Ee [a,*,1], René Sitters [a,b]

[a] Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, the Netherlands
[b] Centrum voor Wiskunde en Informatica (CWI), Science Park 123, 1098 XG Amsterdam, the Netherlands

A R T I C L E I N F O

Article history:
Received 29 September 2017
Received in revised form 9 April 2018
Accepted 11 April 2018
Available online xxxx
Communicated by J. Díaz

Keywords:
Canadian traveler problem
Graph search problem
Approximation algorithms
Computational complexity
Routing under uncertainty

A B S T R A C T

In the Canadian traveler problem, we are given an edge weighted graph with two specified vertices s and t and a probability distribution over the edges that tells which edges are present. The goal is to minimize the expected length of a walk from s to t. However, we only get to know whether an edge is active the moment we visit one of its incident vertices. Under the assumption that the edges are active independently, we show NP-hardness on series-parallel graphs and give results on the adaptivity gap. We further show that this problem is NP-hard on disjoint-path graphs and cactus graphs when the distribution is given by a list of scenarios. We also consider a special case called the multi-target graph search problem. In this problem, we are given a probability distribution over subsets of vertices. The distribution specifies which set of vertices has targets. The goal is to minimize the expected length of the walk until finding a target. For the independent decision model, we show that the problem is NP-hard on trees and give a $(3.59+\epsilon)$-approximation for trees and a $(14.4+\epsilon)$-approximation for general metrics. For the scenario model, we show NP-hardness on star graphs.

1. Introduction

In this paper, we consider the *Canadian traveler problem* (CTP). Here, we are given an edge-weighted graph $G=(V,E)$ with two specified vertices s and t, and a probability distribution $p: 2^E \to [0,1]\cap\mathbb{Q}$. This function gives for each $A \subseteq E$ the probability that A is the active set of edges. Only the active edges are present. We need to construct a walk from s to t. However, we only know whether an edge is present when we visit one if its incident vertices. The problem is to find a policy that minimizes the expected length of our walk. Here, a policy may use the observed realizations as input to decide where to go next. We consider the problem in the independent decision model and in the scenario model. In the independent decision model, each edge has a probability of being present and the event of an edge being present is independent of the other edges. In the scenario model, we are given an explicit list of scenarios $\mathcal{S}$, where each scenario describes which edges are present. The probability distribution p is known in advance. We assume that the edge weights are integer.

It was shown in [20] that the decision version of CTP is NP-complete in the scenario model. They also showed that the problem is polynomially solvable if the number of scenarios is bounded by a constant. In the independent decision model,

* Corresponding author.
E-mail address: M.v.Ee.01@mindef.nl (M. van Ee).

[1] Present address: Netherlands Defence Academy, Het Nieuwe Diep 8, 1781 AC Den Helder, The Netherlands.

https://doi.org/10.1016/j.tcs.2018.04.022

the decision version of CTP is PSPACE-complete [10] and it is #P-hard to compute the expected length [19,20]. It is even not possible to describe an optimal policy, unless PSPACE $\subseteq$ P/poly [24]. On the other hand, the problem is polynomially solvable on disjoint-path graphs [5]. Until now, the computational complexity of the problem was still open on series-parallel graphs. The complexity of CTP on this class of graphs was mentioned as one of the major open problems in [18]. Fried [9] considered CTP on a subset of graphs which become trees after deleting t. He conjectured that CTP is intractable in this case. It is easy to see that the class of graphs considered by Fried is a subset of the class of series-parallel graphs. A consequence of our work is that CTP is indeed NP-hard in both cases.

The problem also has an adversarial version, i.e., there is no probability distribution but there is an adversary that chooses the edges that are present. In this problem, we compare the length of the walk with the offline optimum, i.e., the optimal solution if we had full knowledge of the edges that are present. The worst-case ratio between these values is called the competitive ratio. It is reasonable to consider the restriction that at most k edges fail. This variant is called *k-CTP* and was introduced by Bar-Noy and Schieber [3]. It was shown in [25] that the Backtrack-algorithm that repeatedly chooses the shortest path and returns when the path is blocked, is $2k+1$-competitive. Westphal [25] also showed that no algorithm can beat this bound. In our setting, the Backtrack-algorithm is an $O(n)$-approximation in the independent decision and in the scenario model, since we have full information about the graph after at most n turns. The Backtrack-algorithm is also an $O(|\mathcal{S}|)$-approximation in the scenario model, since we know which scenario is active after at most $|\mathcal{S}|$ turns. The main open problem is to improve these approximability results.

An important modeling issue, when considering approximation algorithms, is how to deal with an st-disconnected graph. In this case, no walk can reach t. In the independent decision model, this is usually solved by adding an edge from s to t that is present with probability one and has an arbitrary large length. This modeling choice does not influence the computational complexity of the problem. However, it does influence the analysis of approximation algorithms. To get a sensible model from this perspective, we choose to minimize the conditional expectation of the length given that G contains an st-path. Equivalently, the value of a walk is zero whenever the active set of edges induces an st-disconnected graph. This way, we get an objective value equal to the conditional expected length times the probability of having an st-connected graph. We will use this objective function in this paper. For the independent decision model this is a stronger formulation in the sense that if we have an α-approximation in this formulation, we also have an α-approximation in the former one, but not vice versa. In the scenario model we can simply avoid this issue by deleting scenarios that induce an st-disconnected graph and normalize the remaining probabilities.

Before giving results on CTP, we consider its relation with the *multi-target graph search problem* (multi-target GSP), a generalization of the *graph search problem* (GSP) [16,2]. The GSP is formally defined as follows.

Definition 1. Suppose we are given an edge-weighted graph $G=(V,E)$ with root s and probability p_v for each vertex v, such that $\sum_v p_v = 1$. For a given path starting at s, the latency of v, C_v, is defined as the length of the subpath from s to v. In the graph search problem (GSP), the goal is to find a path on all vertices of V that starts in s and minimizes $\sum_v p_v C_v$, i.e., the total weighted latency.

The probability distribution in the GSP specifies the probability that the target is at the corresponding vertex. Hence, the goal in the GSP is to find a walk along the vertices that minimizes the expected length until finding the target. When all probabilities are equal (or polynomially bounded), the problem reduces to the *traveling repairman problem* (TRP), also known as the *minimum latency problem*. The TRP is formally defined as follows.

Definition 2. Suppose we are given an edge-weighted graph $G=(V,E)$ with root s. For a given path starting at s, the latency of v, C_v, is defined as the length of the subpath from s to v. In the traveling repairman problem (TRP), the goal is to find a path on all vertices of V that starts in s and minimizes $\sum_v C_v$, i.e., the total latency.

On general metrics, the current best approximation guarantee for TRP is 3.59 [6]. The problem is even NP-complete on weighted trees [22], but admits a PTAS in the Euclidean plane and on weighted trees [23]. In [2], the authors gave a 40-approximation for GSP by using similar techniques.

Here, we generalize the GSP to the case where there can be multiple targets. For this, we again consider the independent decision model and the scenario model. In the independent decision model, each vertex has a probability of having a target and the event of having a target at a vertex is independent of the presence of targets at other vertices. In the scenario model, we are given an explicit list of scenarios $\mathcal{S}$, where each scenario describes at what vertices a target is present. Now, we want to minimize the expected length of the walk until a target has been found, given that there is at least one target. The following theorem states how the multi-target GSP is related to the CTP. Recall that our objective value is equal to the conditional expected length times the probability of having an st-connected graph.

Theorem 1. *The multi-target GSP is equivalent to a special case of the CTP.*

Proof. We prove the theorem for the independent decision model. The proof for the scenario model is similar and omitted here. Given an instance of multi-target GSP, i.e., an edge-weighted graph $G=(V,E)$ with root s and a probability p_v for

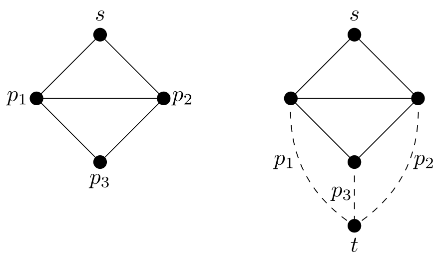

Fig. 1. An instance of multi-target GSP (left) and the same instance shown as a special case of CTP (right). Here, dashed edges have length zero and solid edges have length one. The p_e next to a dashed edge denotes the probability of this edge; solid edges have probability one.

each vertex v, create an instance of CTP by copying the graph and taking s as the start vertex. Add a new vertex t and add edges between t and all vertices in $V \setminus \{s\}$. Edge (v, t) has weight zero and is active with probability p_v. All edges in E are active with probability one. The reduction is illustrated in Fig. 1. Now, there is walk in the multi-target GSP-instance of expected length at most B if and only if there is a walk in the created CTP-instance of expected length at most B. $\square$

We show that we can use the techniques from TRP-algorithms [4,12] to obtain constant-factor approximations for the multi-target GSP in the independent decision model. More precisely, we give a $(3.59+\epsilon)$-approximation for tree metrics and a $(14.4+\epsilon)$-approximation for general metrics. For the scenario model, we show that the problem is NP-complete on star graphs, even when each scenario contains only two targets. For CTP, we investigate the adaptivity gap in the independent decision model. This gap measures the loss when restricting ourselves to non-adaptive solutions. We show that this is strictly greater than one on trees.

We also show that multi-target GSP in the independent decision model is NP-hard on trees. From this fact it follows that CTP is NP-hard on series-parallel graphs. For the scenario model, we strengthen the result from [20], by showing that CTP is even NP-hard on disjoint-path graphs and cactus graphs. The CTP in the independent decision model is solvable on these graph types [5]. The NP-hardness of CTP on general graphs in the scenario model easily follows from the hardness of GSP and the relation between these two problems, as stated in Theorem 1. In the next section, we will discuss results on the independent decision model and in Section 3 we will treat all results concerning the scenario model.

2. Independent decision model

2.1. Multi-target graph search

We start by showing that multi-target GSP is strongly NP-hard on trees by a reduction from TRP on trees [22]. The idea is to assign a probability p to each vertex, where p is small. We then prove that it is sufficient to choose p only polynomially small. Remember that we defined the objective function as the conditional expected length times the probability of having an st-connected graph.

Theorem 2. *Multi-target GSP in the independent decision model is strongly NP-hard on trees.*

Proof. We reduce from TRP on trees where the edge lengths are polynomially bounded integers. As shown in [22], this problem is strongly NP-hard. Hence, we assume that the sum W of the edge lengths is bounded by $W \le n^\ell$ for some constant ℓ. Given such a TRP instance I, we create an instance I' of multi-target GSP by assigning a probability $p = \frac{1}{2n^{\ell+2}}$ to each vertex.

We prove that for any positive integer K, there exists a TRP solution for I with total latency at most K if and only if there exists a solution for I' with expected path length at most pK. Note that the solution set is the same for both problems, namely a walk in the graph that starts at root s and visits all vertices. Consider any solution and let C^{GSP} be the expected path length until finding a target and let C^{TRP} be the total latency of the vertices. Let C_v be the latency of vertex v. Hence, $C^{TRP} = \sum_{v \in V} C_v$. Further,

$$\begin{aligned} C^{GSP} &= pC_1 + (1-p)pC_2 + \ldots + (1-p)^{n-1}pC_n \\ &= p\sum_{v\in V} C_v - p\sum_{k=1}^{n-1}(1-(1-p)^k)C_{k+1} \\ &= pC^{TRP} - p\sum_{k=1}^{n-1}(1-(1-p)^k)C_{k+1}. \end{aligned} \tag{1}$$

From the equation above we that $C^{GSP} < pC^{TRP}$ for $0 < p < 1$. Hence, if $C^{TRP} \le K$ then $C^{GSP} < pK$.

Please cite this article in press as: M. van Ee, R. Sitters, Approximation and complexity of multi-target graph search and the Canadian traveler problem, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.04.022

Vice versa, if $\mathcal{C}^{GSP} \le pK$ then

$$\mathcal{C}^{TRP} \le K + \sum_{k=1}^{n-1} (1-(1-p)^k) C_{k+1}. \tag{2}$$

Next, we show that the summation in the right hand side of (2) is strictly less than 1. Then $\mathcal{C}^{TRP} < K+1$ and since $\mathcal{C}^{TRP}$ is integer, this implies $\mathcal{C}^{TRP} \le K$.

Note that in general, $1 - x/2 > e^{-x} > (1-1/y)^{xy}$ for $0 < x < 1 < y$. We take $x = 2p = \frac{1}{n^{\ell+2}}$ and $y = n^{\ell+1}$ and get

$$1 - p > \left(1 - \frac{1}{n^{\ell+1}}\right)^{\frac{n^{\ell+1}}{n^{\ell+2}}} = \left(1 - \frac{1}{n^{\ell+1}}\right)^{1/n}.$$

Hence, $1-(1-p)^n < 1/n^{\ell+1}$. Next, we use $C_k \le n^\ell$ for all k to bound the summation in (2).

$$\begin{aligned}\sum_{k=1}^{n-1} (1-(1-p)^k) C_{k+1} &\le \sum_{k=1}^{n-1} (1-(1-p)^n) n^\ell \\ &< n^{\ell+1}(1-(1-p)^n) \\ &< n^{\ell+1}(1/n^{\ell+1})) = 1. \quad \square\end{aligned}$$

The proof above also shows that multi-target GSP on general graphs is at least as hard to approximate as TRP. Since TRP has no constant-factor approximation algorithm if the distances violate the triangle inequality [21], we assume in the remainder of this section that the distances form a metric space. To get approximability results, we take the general approach in TRP-literature. For this, we define the following problem.

Definition 3. Suppose we are given an edge-weighted graph $G = (V, E)$ with root s, probability p_v for each vertex v and a quota P. In the probability quota TSP (PQ-TSP), the goal is to find a set $X \ni s$ and a tour on X such that $1 - \prod_{v \in X}(1 - p_v) \ge P$, i.e., the probability of finding a target in X is at least P, and the tour is of minimum length.

A β-approximation algorithm for PQ-TSP will find, for any given probability P, a tour of length at most βL and probability at least P, if there exists a tour of length L and probability at least P. That means, the approximation is on the length and not on the probability. Theorem 3 states that any β-approximation, by this definition, implies an approximation algorithm for multi-target GSP. Our algorithm uses a slightly different definition of a β-approximation though. Given a length bound L, the algorithm finds a tour of length at most βL and probability at least P if there exists a tour of length L and probability at least P. Hence, we assume that L is given instead of P but note that the approximation is still on the length of the tour. Given a β-approximation algorithm for PQ-TSP one can turn it into a β-approximation algorithm with this alternative definition by applying binary search on P. That means, given L, we find the largest value P such that the length of the tour returned by algorithm is at most βL. If the probabilities are polynomially bounded then only a polynomial number of iterations is needed.

Given a tour T, we refer to the probability that T has at least one target as *the probability* of T. Our algorithm and analysis is based on the algorithm for the minimum latency problem by Goemans and Kleinberg [12]. For $i \ge 1$, let $L_i = 2\gamma^{i+u}$, where u is uniformly distributed on $[0, 1]$ and $\gamma > 1$ is a constant. For $i = 0, 1, 2, \ldots, \kappa$ use the procedure above to obtain a tour T_i of length at most βL_i for which the probability is at least that of any tour of length at most L_i. Here, κ is the minimum i for which T_i contains all points v with $p_v > 0$. For each tour, choose the direction, say left or right, uniformly at random. We concatenate these tours T_i, where already visited vertices are shortcutted. We will show that if we take γ to be the root of $x \ln(x) = x + 1$, which is approximately 3.59, we get a 3.59β-approximation for multi-target GSP.

Lemma 1. *Let T be any tour starting and ending in s. Given that there is at least one target on T, the expected time between starting T and finding a target is at most $|T|/2$ if we take the direction uniformly at random, where $|T|$ denotes the length of T.*

Proof. If there is at least one target on T then if we sum the time until finding a target of both directions of the tour, we get at most $|T|$. Hence, if we take the direction uniformly at random, the expected time until finding a target is at most $|T|/2$. $\square$

We shall use Lemma 1 in the following way. Consider the tours $T_1, T_2, \ldots$ constructed by the algorithm. Assume that for some realization of targets, the first target is found on tour T_i. Then for our analysis we shall ignore the randomness of the direction and assume that the target is found half way tour T_i.

Theorem 3. *Given a β-approximation for PQ-TSP, our algorithm is a randomized 3.59β-approximation for the multi-target GSP, when the edge weights satisfy the triangle inequality and probabilities are polynomially bounded.*

Proof. Fix some instance for the multi-target GSP and let $\tilde{p} = 1 - \prod_{v=1}^{n}(1 - p_v)$ be the probability that there is at least one target. Consider some corresponding optimal solution (a walk starting from s) and let $\textsc{Opt}$ be its value. For $p \in [0, \tilde{p}]$, let C_p^* be the length of the optimal walk until probability p is reached. Then, $\textsc{Opt} = \int_0^{\tilde{p}} C_p^* dp$.

For the moment, fix some $p \in [0, \tilde{p}]$. Remember that the algorithm first takes $u \in [0, 1]$ uniformly at random and then finds tours $T_1, \ldots, T_\kappa$. Let ℓ be the smallest index such that $L_\ell \geq 2C_p^*$. Then, the probability of T_ℓ is at least p since there is a tour of length at most $2C_p^*$ and of probability at least p, namely, follow the optimal walk until time C_p^* and then return to s. Further, note that ℓ is a random variable depending on u and

$$\mathbb{E}_u[L_\ell] = \mathbb{E}_u[\gamma^u]2C_p^* = \frac{\gamma - 1}{\ln\gamma}2C_p^*. \tag{3}$$

Given the value $u \in [0, 1]$, let $C_p^{\textsc{Alg},u}$ be the length of the algorithm's walk until probability p is reached. Given Lemma 1 and the discussion thereafter, we may assume that probability p is reached latest half way tour T_ℓ. Hence,

$$\begin{aligned}
C_p^{\textsc{Alg},u} &\leq \sum_{i=1}^{\ell-1}|T_i| + |T_\ell|/2 \\
&\leq \beta\left(\sum_{i=1}^{\ell-1}L_i + L_\ell/2\right) \\
&= \beta\left(\frac{L_\ell - 1}{\gamma - 1} + L_\ell/2\right) \\
&< \beta\left(\frac{L_\ell}{\gamma - 1} + L_\ell/2\right) \\
&= \beta\left(\frac{\gamma + 1}{2(\gamma - 1)}\right)L_\ell.
\end{aligned}$$

From (3), we get that

$$\mathbb{E}_u[C_p^{\textsc{Alg},u}] \leq \beta\left(\frac{\gamma + 1}{2(\gamma - 1)}\right)\mathbb{E}_u[L_\ell] = \beta\frac{\gamma + 1}{\ln\gamma}C_p^*.$$

For given u, let $\textsc{Alg}(u)$ be the expected length of the algorithms walk until the first target. Then,

$$\textsc{Alg}(u) = \int_0^{\tilde{p}} C_p^{\textsc{Alg},u} dp.$$

Let $\textsc{Alg} = \mathbb{E}_u[\textsc{Alg}(u)]$. Then,

$$\textsc{Alg} = \mathbb{E}_u[\int_0^{\tilde{p}} C_p^{\textsc{Alg},u} dp] = \int_0^{\tilde{p}} \mathbb{E}_u[C_p^{\textsc{Alg},u}]dp \leq \beta\frac{\gamma + 1}{\ln\gamma}\int_0^{\tilde{p}} C_p^* dp = \beta\frac{\gamma + 1}{\ln\gamma}\textsc{Opt}.$$

Optimizing over γ gives $\gamma = 3.59$, the unique root of $x \ln x = x + 1$, which gives $\textsc{Alg} \leq 3.59\beta\textsc{Opt}$. We can derandomize this algorithm by using techniques from [12]. □

We now show that PQ-TSP is NP-hard, even on star graphs. Also, we obtain an FPTAS for this problem on tree metrics. In the following reduction, we reduce from Partition. In this problem, we are given n integers $a_1, \ldots, a_n$, and the question is whether there exists a set $X \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in X} a_i = \frac{1}{2}\sum_{i=1}^{n} a_i$. This problem is weakly NP-hard [15].

Theorem 4. *PQ-TSP is weakly NP-hard on star graphs.*

Proof. Given is an instance of Partition. Let $B = \frac{1}{2}\sum_{i=1}^{n} a_i$. Construct a star graph with n leafs, where each leaf is associated with an integer from the Partition-instance. Assign a weight of a_i to the edge connecting the root with the leaf corresponding to integer a_i, and set the probability of finding a target at this leaf equal to a_i/K, where $K > 2^n(\max_i a_i)^2$. We will show that there is a tour of length at most $2B$ with probability strictly larger than $(B - 1)/K$ if and only if there exists a solution X for the Partition instance. For any $X \subseteq \{1, \ldots, n\}$ let $P(X)$ be the probability that there is at least one target in X.

Suppose we have a solution X for the Partition instance. The tour on the vertices corresponding to X has length $2B$ and probability

$$P(X) = 1 - \prod_{i \in X}(1 - a_i/K) = \sum_{i \in X} a_i/K - \sum_{X' \subseteq X, |X'| \geq 2} (-1)^{|X'|} \prod_{i \in X'} (a_i/K)$$

Note that for any $X' \subseteq X$ with $|X'| \geq 2$ we have $\prod_{i \in X'}(a_i/K) \leq (\max_i a_i)^2/K^2$ since $a_i/K < 1$. Hence,

$$P(X) \geq \sum_{i \in X} a_i/K - 2^n (\max_i a_i)^2/K^2 > B/K - K/K^2 = (B-1)/K.$$

If, on the other hand, there is no solution to the Partition instance, then for any $X \subseteq \{1, \ldots, n\}$ either $\sum_{i \in X} a_i > B$ or $\sum_{i \in X} a_i < B$. In the first case, any tour on X has length larger than $2B$. In the second case, the probability will be

$$P(X) = 1 - \prod_{i \in X}(1 - a_i/K) \leq \sum_{i \in X} a_i/K \leq (B-1)/K. \quad \square$$

Theorem 5. *There is an FPTAS for PQ-TSP on tree metrics.*

Proof. We first give a dynamic programming algorithm that, for each tour length, finds the solution with maximum probability. This algorithm runs in pseudopolynomial time. Then, we show how to round the lengths of the edges such that we lose at most a factor $(1+\epsilon)$ in the length of the tour. Note that the solution is a traversal of a subtree, hence it is sufficient to find a tree.

First, we may assume that the tree is binary and that all positive probabilities are on the leafs. Note that this is without loss of generality since if $p_v > 0$ and v is not a leaf then we add a new edge with length 0 and move the probability to the leaf. Next, we can turn the tree into a binary tree by adding edges of length zero.

We also guess the farthest vertex from the root visited by our tour, and remove all vertices at a larger distance from the root. Call the corresponding distance D and note that, given that our guess is correct, we have $\text{Opt} \geq D$, where Opt is the value of the optimal solution of PQ-TSP. Our dynamic programming formulation now has a state for each vertex-length pair, denoted by (v, L). We define $f(v, L)$ as the maximum probability of finding a target in the subtree rooted at v using a tree of weight at most L. By definition, $f(v, L) = 0$ whenever $L < 0$. If we denote the left and right child of v as $\hat{v}$ and $\tilde{v}$ respectively and if we use $c(\cdot, \cdot)$ for the edge lengths, we can compute this value as follows.

$$f(v, L) = \max_{0 \leq \lambda \leq L} \{1 - (1 - f(\hat{v}, \lambda - c(v, \hat{v})))(1 - f(\tilde{v}, L - \lambda - c(v, \tilde{v})))\}$$

The optimal solution can be computed by applying binary search on $f(s, L)$ for different L. For each guess of the farthest vertex, there are at most $n^2 D$ states, which can each be evaluated in $O(nD)$ time. The binary search procedure takes $O(\log(nD)) = O(\log n + \log D)$ time. Hence, for each guess, this algorithm runs in $O(n^3 D^2 (\log n + \log D))$. To get a polynomial time algorithm, we pick $\epsilon > 0$ and we round each of the edge lengths up to its nearest multiple of $\epsilon D/n$. This way, we lose at most $\epsilon D \leq \epsilon \text{Opt}$ and hence a factor $(1+\epsilon)$ in the length of the tour, but we have reduced the running time to $O(n^5 (\log n + \log D)/\epsilon^2)$. Since the total running time is now $O(n^6 (\log n + \log D)/\epsilon^2)$, we have obtained a fully polynomial time approximation scheme for PQ-TSP on tree metrics. $\square$

Note that to obtain approximation results for multi-target GSP using Theorem 3, it is sufficient to approximate the PQ-TSP where we assume that the length L is given instead of the probability P. The proof above shows that for a given L, we can find a tour of length at most $(1+\epsilon)L$ and probability P if there exists a tour of length at most L and probability P in $O(n^6/\epsilon^2)$ time.

Corollary 1. *There is a $(3.59 + \epsilon)$-approximation for multi-target GSP on tree metrics when probabilities are polynomially bounded.*

Before discussing the approximability of multi-target GSP on general metrics, we first generalize PQ-TSP. We define the *polymatroid quota TSP* as follows. Here, a function q is polymatroid [17] if it satisfies

- $q(\emptyset) = 0$,
- $q(I) \leq q(J)$ for all $I \subseteq J$,
- $q(I \cup \{z\}) - q(I) \geq q(J \cup \{z\}) - q(J)$ for all $I \subseteq J$ and all $z \notin J$.

Definition 4. Suppose we are given an edge-weighted graph $G = (V, E)$ with root s, polymatroid function $q : 2^V \to \mathbb{Q}$ and a quota Q. In the polymatroid quota TSP, the goal is to find a set $X \ni s$ and a tour on X such that $q(\bar{X}) \leq Q$, where $\bar{X}$ is the complement of X, and the tour is of minimum length.

PQ-TSP is a special case of polymatroid quota TSP. This can be seen by taking $q(X) = 1 - \prod_{v \in X}(1 - p_v)$ and setting $Q = 1 - \frac{\prod_v (1-p_v)}{1-P}$. Note that it also generalizes k-TSP [11], the problem of finding a tour on k vertices with minimum length, since $q(X) = |X|$ is polymatroid and $|X| \geq k$ is equivalent to $|\bar{X}| \leq n - k$.

To get approximation results for polymatroid quota TSP, we use results for the *prize-collecting TSP with polymatroid penalties*. In this problem, we can decide not to visit certain vertices. The penalty cost of not visiting these vertices is determined by a polymatroid function π. Now, we have to find a tour on a subset X of the vertices such that the sum of tour cost $L(X)$ and penalty cost $\pi(\bar{X})$ is minimized.

When $\pi(\bar{X}) = \sum_{i\in\bar{X}} \pi_i$, where π_i is the penalty paid for not visiting a vertex, the problem is known as the *prize-collecting TSP*. For this problem, Goemans and Williamson [13] showed that their algorithm produces a solution that visits X such that

$$L(X) + \pi(\bar{X}) \le 2 \sum_{X\subseteq V\setminus\{s\}} y_X, \tag{4}$$

where $\sum_X y_X$ is the objective value of the dual of the LP-relaxation of the prize-collecting TSP. Goemans and Kleinberg [12] observe that the proof of [13] even shows that

$$L(X) + 2\pi(\bar{X}) \le 2 \sum_{X\subseteq V\setminus\{s\}} y_X,$$

if $\pi(\bar{X}) = \sum_{i\in\bar{X}} \pi_i$. On the other hand, it was observed in [14] that the Goemans–Williamson algorithm [13] is a 2-approximation for prize-collecting Steiner tree with polymatroid penalties. One can extend this result to the prize-collecting TSP with polymatroid penalties. These results hold since (4) also holds if π is a polymatroid function. Moreover, we can easily extend this result to the following lemma, again using the proof of [13].

Lemma 2. *The Goemans–Williamson algorithm [13] produces a solution that visits set X with the property*

$$L(X) + 2\pi(\bar{X}) \le 2 \sum_{X\subseteq V\setminus\{s\}} y_X,$$

where $\sum_X y_X$ is the objective value of the dual of the LP-relaxation of the prize-collecting TSP with polymatroid penalties.

This lemma can be used to obtain a 5-approximation for polymatroid quota TSP. This is done using the approach demonstrated in [7]. The authors observe that the dual of the prize-collecting Steiner tree problem and the dual of the Lagrangian relaxation of the k-minimum spanning tree problem, i.e., the problem of finding a minimum cost tree spanning at least k vertices, are very similar. This relation easily extends to TSP-versions of both problems and to quota TSP, PQ-TSP, and polymatroid quota TSP with their corresponding prize-collecting versions. As observed by [7], we can use techniques from [1] to improve this factor to $4+\epsilon$. Hence, we have a $(4+\epsilon)$-approximation for PQ-TSP on general metrics.

Since the proofs used by Goemans and Williamson [13] and Chudak et al. [7] are extensive and our contribution follows by straightforward adjustments, we omitted these proofs from this paper.

Now, since the binary search procedure and the algorithm for the subproblem take polynomial time, we apply Theorem 3 to obtain a $(14.4+\epsilon)$-approximation for multi-target GSP.

Corollary 2. *There is a $(14.4+\epsilon)$-approximation for multi-target GSP when the probabilities are polynomially bounded and the edge weights satisfy the triangle inequality.*

2.2. Canadian traveler problem

In this subsection, we discuss the complexity and approximability of CTP in the independent decision model. For the complexity, it follows from Theorem 1 and 2 that it is NP-hard on series-parallel graphs.

Corollary 3. *The CTP in the independent decision model is NP-hard on series-parallel graphs.*

Since CTP in the independent decision model is easy on disjoint-path graphs, this basically settles the computational complexity of the problem. For the approximability, we look at the power of being adaptive. To investigate this, we define non-adaptive policies and the adaptivity gap. In order to have a correct definition, we assume that all edges incident to t have weight zero. This assumption is without loss of generality since we can extend each edge to t with a new edge of weight zero and probability one.

Definition 5. We say that a policy is non-adaptive if it has a fixed ordering of the vertices and it always tries to reach the first unvisited vertex in this order. If the vertex is disconnected with the root, i.e., we cannot reach it, the vertex is discarded and the next vertex in the ordering is considered. If it reaches one of the vertices adjacent to t and the edge to t is present, it visits t next.

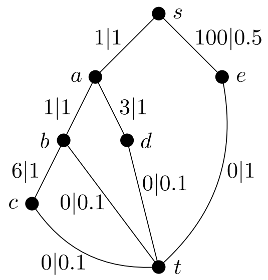

Fig. 2. Instance with adaptivity gap greater than 1. Labels denote "length|probability".

Definition 6. Let $\textsc{Opt}(I)$ and $\text{NA}(I)$ be the value of the optimal solution and the value of the optimal non-adaptive policy for instance I of CTP respectively. Then, the adaptivity gap is defined as

$$\sup_I \frac{\text{NA}(I)}{\textsc{Opt}(I)}.$$

On trees, i.e., when the graph without t is a tree, a non-adaptive policy is just a permutation of the vertices, where we can skip a subtree if the top edge of the subtree is not present. The instance in Fig. 2 shows that we have to be adaptive in order to be optimal.

Theorem 6. *The adaptivity gap of CTP in the independent decision model on trees is greater than 1.*

Proof. Consider the instance in Fig. 2. There are three non-adaptive solutions that have to be considered. These are $Y_1 = (s, e, a, b, d, c)$, $Y_2 = (s, a, b, d, c, e)$ and $Y_3 = (s, a, d, b, c, e)$. We get the following expected values, denoted by $C(\cdot)$.

$$C(Y_1) = 0.5 \cdot 100 + 0.5(0.1 \cdot 2 + 0.1 \cdot 0.9 \cdot 6 + 0.1 \cdot 0.9^2 \cdot 16) \approx 51.018$$
$$C(Y_2) = 0.1 \cdot 2 + 0.1 \cdot 0.9 \cdot 6 + 0.1 \cdot 0.9^2 \cdot 16 + 0.9^3 \cdot 0.5 \cdot 124 \approx 47.234$$
$$C(Y_3) = 0.1 \cdot 4 + 0.1 \cdot 0.9 \cdot 8 + 0.1 \cdot 0.9^2 \cdot 14 + 0.9^3 \cdot 0.5 \cdot 122 \approx 46.723$$

However, an adaptive solution is allowed to condition on the presence of edge (s, e). If this edge is active, it will do (s, a, b, c, d, e). Otherwise, it will do (s, a, b, d, c). This gives the following value of adaptive solution Ad.

$$\begin{aligned} C(Ad) = {} & 0.5(0.1 \cdot 4 + 0.1 \cdot 0.9 \cdot 8 + 0.1 \cdot 0.9^2 \cdot 14 + 0.9^3 \cdot 122) \\ & + 0.5(0.1 \cdot 2 + 0.1 \cdot 0.9 \cdot 6 + 0.1 \cdot 0.9^2 \cdot 16) \approx 46.614 \end{aligned}$$

Hence, the adaptivity gap of independent-CTP on trees is at least 1.0023. □

In order to achieve larger lower bounds, one has to try other approaches than used above by the following reasoning. In the instance in Fig. 2, the optimal TSP-solution for the subgraph induced by $\{a, b, c, d\}$ is (a, d, b, c), while the optimal TRP-solution is (a, b, d, c). The adaptive algorithm is stronger since it can choose which of these solutions to use depending on the presence of edge (s, e). Extending this approach to get better lower bounds will not result in satisfying results. This is because one can approximate TSP and TRP simultaneously. To see this, one could take the algorithm by Goemans and Kleinberg [12] for TRP using Garg's 2-approximation for k-TSP [11] as its subroutine. This algorithm is a 7.2-approximation for TRP, but it is also an 8-approximation for TSP. Hence, in order to achieve significantly larger lower bounds, one has to use more elaborate approaches.

Let us end this section with a discussion on Definition 5. On trees, it perfectly captures the intuition of a non-adaptive policy: a permutation of the vertices. It is debatable what a correct description of a non-adaptive policy is on general graphs. Definition 5 says that a non-adaptive policy will always try to reach the next unvisited vertex, but it is not specified how to do this. A policy could for example use the shortest path with respect to edge lengths to the next vertex. It is also allowed to use the path that maximizes the probability of reaching the next vertex. Here, we want to emphasize that it is arguable what the correct definition of a non-adaptive policy is.

3. Scenario model

We now consider multi-target GSP and CTP in the scenario model. We first discuss the complexity of the former problem on star graphs, and show that it is NP-hard when each scenario contains only two targets. As a consequence, by Theorem 1, CTP is NP-hard on disjoint-path graphs. Finally, we show that CTP in the scenario model on cactus graphs (graphs with the

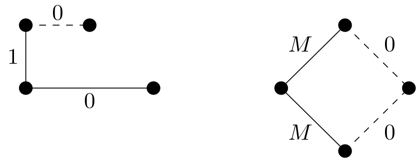

Fig. 3. Gadget Z_1 (left) and Z_2 (right), where dashed edges are uncertain.

property that each edge is in at most one cycle) is also NP-hard. At this point, we would like to mention that the CTP in the scenario model has similar features with CTP with dependencies [10] and CTP with remote sensing [5,10]. In all these problems, it might be beneficial to explore the state of the graph before proceeding along a path. This will be used when proving that CTP in the scenario model on cactus graphs is NP-hard.

3.1. Multi-target graph search

First observe that multi-target GSP in the scenario model is a generalization of GSP, since this is the case when $|S| = 1$ for all $S \in \mathcal{S}$. On star graphs, this problem is easily solvable by visiting the leafs in non-decreasing order of the ratio of the probability and the length of the edge to the leaf.

Let us now discuss the case where $|S| = 2$ for all $S \in \mathcal{S}$, called the *two-target GSP*. We will give a reduction from the *min sum vertex cover problem* (MSVC). In this problem, we are given a graph $G = (V, E)$ and we have to find a linear ordering of the vertices, i.e., a bijection $f : V \to \{1, \ldots, n\}$. The cover time of edge (u, v) is defined as the minimum of $f(u)$ and $f(v)$. The goal is to construct f such that the sum of cover times of the edges is minimized. In [8], it was shown that the problem is NP-hard.

Theorem 7. *Two-target GSP is NP-hard on star graphs.*

Proof. Given an instance of MSVC, i.e., a graph $G = (V, E)$, we create the following instance for two-target GSP. First, construct an unweighted star graph with $|V|$ leafs. Secondly, we create a scenario $\{u, v\}$ for each edge $(u, v) \in E$. Each scenario has probability $1/|E|$. Now, we will show that there is a solution for MSVC in the original instance with value z if and only if there is a solution for two-target GSP in the created instance of value $\frac{1}{|E|}(2z - |E|)$.

We can use the linear ordering as our tour for our instance of two-target GSP and vice versa. Now, we have found the target after walking $2h - 1$ distance (and not before $2h - 1$) if the active scenario corresponds to an edge that is covered after h steps in the ordering. Hence, there is a solution for MSVC in the original instance with value z if and only if there is a solution for two-target GSP in the created instance of value

$$\frac{1}{|E|} \sum_{(u,v) \in E} (2 \min\{f(u), f(v)\} - 1) = \frac{1}{|E|}(2z - |E|). \quad \square$$

3.2. Canadian traveler problem

For the scenario model, we can prove NP-hardness for two extreme cases of series-parallel graphs, namely disjoint-path graphs and cactus graphs (graphs with the property that each edge is in at most one cycle). That this is true for the former graph type follows immediately from Theorem 1 and 7, since adding edges from the leafs of the star graph to a new vertex t leads to a disjoint-path graph ($K_{2,n}$).

Corollary 4. *CTP in the scenario model is NP-hard on disjoint-path graphs.*

For cactus graphs, we need to do some more work. We reduce from *exact cover by 3-sets* (X3C) [15]. In this problem, we are given $3q$ elements and m sets, each containing three distinct elements. The question is whether there are q sets such that all elements are covered.

Theorem 8. *CTP in the scenario model is NP-hard on cactus graphs.*

Proof. Starting from an instance of X3C, we construct the following graph. For this, we need the gadgets in Fig. 3, denoted by gadget Z_1 and gadget Z_2.

Assume without loss of generality that $3q + 4$ is a power of 2. We take $m + 3q + 3$ times gadget Z_1 and concatenate these. This is done by identifying the rightmost vertex of a Z_1-gadget to the leftmost vertex of the next Z_1-gadget. The first m gadgets correspond to sets, the next $3q$ gadgets to elements, and the final three to dummies. Take the leftmost vertex of the first gadget as s. Then, take $\log_2 \sigma$ times gadget Z_2, where $\sigma = 3q + 4$. Concatenate these by identifying the rightmost

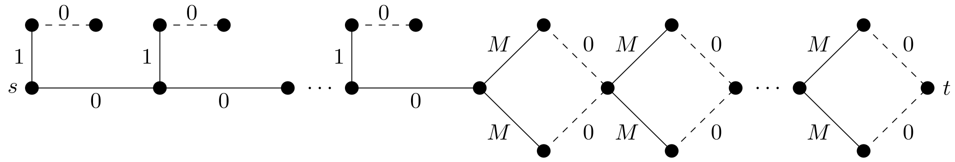

Fig. 4. The constructed graph.

vertex of a Z_2-gadget to the leftmost vertex of the next Z_2-gadget. Then, identify the rightmost vertex of the first chain with the leftmost vertex of the second chain. Finally, we take the rightmost vertex of the second chain as t. Note that the resulting graph (Fig. 4) is a cactus graph.

The idea of the reduction is that we cannot afford to make a mistake in the Z_2-chain, that means, we need to know exactly which scenario is active before reaching the Z_2-chain. Learning which scenario is active is done by moving to the uncertain edges in the Z_1-chain. We say that we ‘try out’ the gadget. If there is an exact cover then learning can be done cheaper than when there is no exact cover.

We define σ scenarios, where each scenario corresponds to an element or to a dummy or neither. All scenarios have equal probability $\frac{1}{\sigma}$. A scenario corresponding to an element contains, next to all certain edges, the uncertain edge in the gadget corresponding to the element and in the gadgets corresponding to sets containing this element. A scenario corresponding to a dummy contains, next to all certain edges, the uncertain edge in the gadget corresponding to that dummy. There is one scenario that corresponds to neither of them. This scenario contains none of the uncertain edges in the first chain of gadgets. Moreover, we assign a unique path in the second chain to each scenario. This can be done as follows. For scenario j, the top edge of gadget z is active (and the bottom edge is not), if the zth digit in the binary representation of $j-1$ is a 1. Otherwise, the bottom edge is available.

Finally, we let B, the bound on the objective value of CTP, equal $M\log_2\sigma + (3q^2+21q+18)/\sigma$. By setting $M > 3q^2 + 21q + 18$, we know we will exceed B if we make guesses in the second chain, since each guess costs at least an extra M/σ. Therefore, one has to learn the scenario in the first chain of gadgets.

Assume there is an exact cover and let Y be a solution. Then, first we try out the gadgets corresponding to sets in Y. If an uncertain edge turns out to be active then we try out either one or two of the three corresponding element-gadgets until we know which scenario is active, and then move to the Z_2-chain. If no gadget from Y is active, we try out the three dummies until either the active one is found or none of them is active, and then move to the Z_2-chain. This policy has an expected length of

$$M\log_2\sigma + \underbrace{\frac{1}{\sigma}(4+6+6)}_{\text{contribution first set}} + \ldots + \underbrace{\frac{1}{\sigma}(2q+2+2q+4+2q+4)}_{\text{contribution } q\text{th set}}$$
$$+ \underbrace{\frac{1}{\sigma}(2q+2+2q+4+2q+6)}_{\text{contribution dummies}} + \frac{1}{\sigma}(2q+6) = B.$$

Now, suppose that we have a yes-instance for CTP, i.e., we have a policy with an expected length of at most B. We need to show that this implies that we have a yes-instance for X3C.

As observed before, we need to learn the scenario on the first part. Moreover, once we see an active set, we will try out either one or two of the corresponding elements and take the active path. In general, the policy looks as follows. We will first try out k sets with three new elements. Then, we will try out ℓ sets with two new elements. Finally, we will try out the remaining elements and dummies. This policy has an expected value of

$$\begin{aligned}
M\log\sigma &+ \frac{1}{\sigma}(4+6+6) + \ldots + \frac{1}{\sigma}(2k+2+2k+4+2k+4) \\
&+ \frac{1}{\sigma}(2(k+1)+2+2(k+1)+2) + \ldots + \frac{1}{\sigma}(2(k+\ell)+2+2(k+\ell)+2) \\
&+ \frac{1}{\sigma}(2(k+\ell+1)+\ldots+2(k+\ell+3q-3k-2\ell+3)) \\
&+ \frac{1}{\sigma}(2(k+\ell+3q-3k-2\ell+3)) \\
= M\log\sigma &+ \frac{1}{\sigma}(9q^2+6k^2+2\ell^2-12qk-6q\ell+6k\ell+27q-6k-4\ell+18) \\
= B &+ \frac{1}{\sigma}(6q^2+6k^2-12qk+2\ell^2+6q-6k-6q\ell+6k\ell-4\ell)
\end{aligned}$$

$$
\begin{aligned}
&= B + \frac{1}{\sigma}(6(q-k)^2 + (6-6\ell)(q-k) + 2\ell^2 - 4\ell) \\
&= B + \frac{1}{\sigma}(2(\sqrt{3}(q-k) - \ell)^2 + 6(q-k) + (4\sqrt{3}-6)\ell(q-k) - 4\ell) \\
&\geq B + \frac{1}{\sigma}(2(\sqrt{3}(q-k) - \ell)^2 + (4\sqrt{3}-6)\ell(q-k)),
\end{aligned}
$$

where the last inequality follows from $\ell \leq \frac{3}{2}(q-k)$, since this is the maximum number of sets with two new elements you can choose after picking k disjoint 3-sets. Now, since $q \geq k$ and $\ell \geq 0$, this policy has expected value greater than or equal to B. Moreover, the expected value of the solution is equal to B if and only if $q = k$ and $\ell = 0$. Hence, we have a yes-instance for X3C. □

4. Conclusion

In this paper, we considered the Canadian traveler problem and the multi-target graph search problem. For the CTP in the independent decision model, we showed NP-hardness on series-parallel graphs. This followed immediately after showing that multi-target GSP in the independent decision model is NP-hard on trees. For the multi-target GSP in the independent decision model we gave a $(3.59 + \epsilon)$-approximation for trees and a $(14.4 + \epsilon)$-approximation on general metrics. We also showed that one should be adaptive in order to be optimal for CTP.

For the scenario model, we gave an NP-hardness proof for two-target GSP on star graphs. As a consequence, CTP in the scenario model is NP-hard on disjoint-path graphs. Finally, we showed that this problem is also NP-complete on cactus graphs.

The main open problem in this field remains the approximability of CTP in the independent decision model. As a start, one could try to prove an upper bound on the adaptivity gap. We conjecture that this is bounded by a small constant on trees. In general, an approach could be to investigate good adaptive algorithms.

Acknowledgement

The authors are supported by the NWO Grant 612.001.215.

References

[1] Sanjeev Arora, George Karakostas, A $2 + \epsilon$ approximation algorithm for the k-MST problem, Math. Program. 107 (3) (2006) 491–504.
[2] Giorgio Ausiello, Stefano Leonardi, Alberto Marchetti-Spaccamela, On salesmen, repairmen, spiders, and other traveling agents, in: G.C. Bongiovanni, G. Gambosi, R. Petreschi (Eds.), Proceedings of the 4th Italian Conference on Algorithms and Complexity, in: Lecture Notes in Comput. Sci., vol. 1767, Springer, 2000, pp. 1–16.
[3] Amotz Bar-Noy, Baruch Schieber, The Canadian traveller problem, in: Proceedings of the 2nd Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, ACM/SIAM, 1991, pp. 261–270.
[4] Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, Madhu Sudan, The minimum latency problem, in: Proceedings of the 26th Annual ACM Symposium on Theory of Computing, ACM, 1994, pp. 163–171.
[5] Zahy Bnaya, Ariel Felner, Solomon Eyal Shimony, Canadian traveler problem with remote sensing, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009, pp. 437–442.
[6] Kamalika Chaudhuri, Brighten Godfrey, Satish Rao, Kunal Talwar, Paths, trees, and minimum latency tours, in: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 2003, pp. 36–45.
[7] Fabian A. Chudak, T. Roughgarden, David P. Williamson, Approximate k-MSTs and k-Steiner trees via the primal-dual method and Lagrangean relaxation, Math. Program. 100 (2) (2004) 411–421.
[8] Uriel Feige, László Lovász, Prasad Tetali, Approximating min sum set cover, Algorithmica 40 (4) (2004) 219–234.
[9] Dror Fried, Theoretical Aspects of the Generalized Canadian Traveler Problem, PhD thesis, Ben-Gurion University of the Negev, 2013.
[10] Dror Fried, Solomon Eyal Shimony, Amit Benbassat, Cenny Wenner, Complexity of Canadian traveler problem variants, Theoret. Comput. Sci. 487 (2013) 1–16.
[11] Naveen Garg, Saving an epsilon: a 2-approximation for the k-MST problem in graphs, in: Proceeding of the 37th Annual ACM Symposium on the Theory of Computing, ACM, 2005, pp. 396–402.
[12] Michel X. Goemans, Jon Kleinberg, An improved approximation ratio for the minimum latency problem, Math. Program. 82 (1–2) (1998) 111–124.
[13] Michel X. Goemans, David P. Williamson, A general approximation technique for constrained forest problems, SIAM J. Comput. 24 (2) (1995) 296–317.
[14] Ara Hayrapetyan, Chaitanya Swamy, Éva Tardos, Network design for information networks, in: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2005, pp. 933–942.
[15] Richard M. Karp, Reducibility among combinatorial problems, in: Complexity of Computer Computations, 1972, pp. 85–103.
[16] Elias Koutsoupias, Christos Papadimitriou, Mihalis Yannakakis, Searching a fixed graph, in: F. Meyer auf der Heide, B. Monien (Eds.), Proceedings of the 23rd International Colloquium on Automata, Languages and Programming, in: Lecture Notes in Comput. Sci., vol. 1099, Springer, 1996, pp. 280–289.
[17] László Lovász, Mathematical Programming the State of the Art, chapter Submodular functions and convexity, Springer, 1983.
[18] Evdokia Nikolova, David R. Karger, Route planning under uncertainty: the Canadian traveller problem, in: Proceedings of the 23rd AAAI Conference on Artificial Intelligence, 2008, pp. 969–974.
[19] Christos H. Papadimitriou, Mihalis Yannakakis, Shortest paths without a map, Theoret. Comput. Sci. 84 (1) (1991) 127–150.
[20] George H. Polychronopoulos, John N. Tsitsiklis, Stochastic shortest path problems with recourse, Networks 27 (2) (1996) 133–143.
[21] Sartaj Sahni, Teofilo Gonzalez, P-complete approximation problems, J. ACM 23 (3) (1976) 555–565.
[22] René Sitters, The minimum latency problem is NP-hard for weighted trees, in: W.J. Cook, A.S. Schulz (Eds.), Proceedings of the 9th International Conference on Integer Programming and Combinatorial Optimization, in: Lecture Notes in Comput. Sci., vol. 2337, Springer, 2002, pp. 230–239.

[23] René Sitters, Polynomial time approximation schemes for the traveling repairman and other minimum latency problems, in: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 604–616.

[24] Cenny Wenner, Hardness Results for the Shortest Path Problem under Partial Observability, Bachelor thesis, Department of computer science, Faculty of science, Lund University, 2009.

[25] Stephan Westphal, A note on the k-Canadian traveller problem, Inform. Process. Lett. 106 (3) (2008) 87–89.