A guide book to make the Canadian Traveller more competitive

Pierre Bergé *,[1], Arpad Rimmel[2], and Joanna Tomasik[2]

[1]LRI, Université Paris-Sud, Université Paris-Saclay, 91405 Orsay Cedex, France
[2]LRI, CentraleSupélec, Université Paris-Saclay, 91405 Orsay Cedex, France

Abstract

The k-Canadian Traveller Problem (k-CTP), known to be PSPACE-complete, generalizes the Shortest Path Problem by considering instances with up to k blocked edges. A blocked edge is revealed when the traveller visits one of its endpoints. The objective is to determine a strategy such that the traveller, guided by it, goes from source s to target t with minimum distance.

Westphal proved that the competitive ratio of any deterministic strategy is not less than $2k+1$. The only family of graphs known until now for which this ratio can be lowered is composed of graphs made up of node-disjoint paths between s and t.

We propose a deterministic polynomial-time strategy, PIVOT-REPOSITION, which can go below $2k+1$ for families of graphs which we identify in polynomial time. Its competitive ratio is $(2-\gamma)\,k+1$ where $0 \leq \gamma \leq 1$ is a parameter depending on the instance. For some graphs for which PIVOT-REPOSITION does not drop below the $2k+1$ limit, we design randomized strategies with a competitive ratio less than $2k+1$.

We establish a topography of k-CTP instances and attribute the quality of appropriate strategies to their subsets. Strategies we designed descend below the $2k+1$ ratio on a set of instances pinpointed in polynomial time.

1 Introduction

The *Canadian Traveller Problem* (CTP) was introduced by Papadimitriou and Yannakakis [11]. Given an undirected weighted graph $G = (V, E, \omega)$ and two nodes $s, t \in V$, the objective is to make a traveller walk from s to t on graph G the most efficiently despite the fact that certain edges may be blocked. The traveller ignores which edges are blocked when he begins his walk and discovers them when he visits an endpoint of these edges. The *k-Canadian Traveller Problem* (k-CTP) is the parameterized version of CTP where an upper bound for the total number of blocked edges is specified. Both CTP and k-CTP are PSPACE-complete [2, 11].

State-of-the-art: Several strategies, which guide the traveller through his walk, have been designed and assessed through the competitive analysis, which is a methodology to evaluate the quality of an online algorithm. The competitive ratio is the ratio between the distance traversed effectively by the traveller and the distance he would have traversed, knowing which edges are blocked before beginning his walk. There are two categories of strategies: deterministic and randomized. Westphal [12] proved that there is no deterministic strategy that can achieve a competitive ratio better than $2k+1$. There are two polynomial-time strategies with this ratio:

*Corresponding author: `Pierre.Berge@lri.fr`

- REPOSITION [12] makes the traveller traverse the shortest (s,t)-path (path from s to t) first. If there is a blockage on this path, the traveller comes back to s after discovering it. Then, the process is restarted on the graph deprived of the identified blocked edge.
- COMPARISON [13] also starts by putting the traveller on the shortest (s,t)-path. If this path is blocked, when the traveller discovers it (we note v the node on which the traveller is standing at this moment), he compares the cost of the shortest (s,t)-path with the cost of the shortest (v,t)-path both found in the graph deprived of the blocked edge discovered. If the (v,t)-path is shorter, the traveller traverses it. Otherwise, he returns to s and traverses the (s,t)-path he identified. The process is restarted each time after a blocked edge discovery.

Randomized strategies, *i.e.* strategies in which choices of direction depend on a random draw, are evaluated by supposing that an oblivious adversary (a person who potentially knows which strategy is used but cannot guess the result of a random draw in advance) is setting the blocked edges. Westphal proved that there is no randomized strategy that achieves a ratio lower than $k+1$ in average [12]. Recently, two randomized strategies have been proposed. Demaine *et al.* [6] designed a strategy with a ratio $\left(1+\frac{\sqrt{2}}{2}\right)k+1$ and executed in time $\mathcal{O}\left(k\mu^2\left|E\right|^2\right)$, where μ is an parameter that can potentially be exponential. This approach concerns exclusively graphs which can be transformed into apex trees using a process they detailed. Then, Bender *et al.* [3] studied the specific case of graphs composed only of node-disjoint (s,t)-paths and proposed a polynomial-time randomized strategy of ratio $k+1$. Their method is a randomized REPOSITION: they assign a probability to each (s,t)-path and execute a draw according to it. The traveller traverses the selected path, returns to s if he is blocked and restarts the process. To the best of our knowledge, no polynomial-time strategy which achieves the competitive ratio smaller than $2k+1$ exists with exception to Bender's method valid for a very restrictive over-mentioned class of instances.

Contributions: We design new strategies and identify for each one a family of instances on which they achieve a ratio smaller than $2k+1$. In summary:

- We propose a *topography* (Section 2.2) to represent the quality of strategies for k-CTP. This image allows us to evaluate the competitive ratio of strategies on families of graphs instead on all instances. Thanks to this chart, we choose the most competitive strategy for a given instance. Conversely, we can say on which instances a given strategy is the most efficient.
- We design a deterministic polynomial-time strategy PIVOT-REPOSITION (Sections 3.1 and 3.2).
- We define a parameterized criterion $C(\gamma)$ (Section 3.3), where $0 \leq \gamma \leq 1$, The competitive ratio of PIVOT-REPOSITION is $(2-\gamma)k+1$ for instances which satisfy criterion $C(\gamma)$. This is the first deterministic strategy proposed with ratio smaller than $2k+1$ for a set of instances identifiable in polynomial time. We prove that no deterministic strategy achieves a competitive ratio better than PIVOT-REPOSITION on instances satisfying $C(\gamma)$.
- We introduce the *hypergraph representation* (Section 4). Thanks to it, we distinguish in polynomial time two types of graphs, *hyperedge-disjoint* and *recursive-partition* graphs. Because PIVOT-REPOSITION does not improve the competitive ratio value for instances containing this kind of graphs, we design specific randomized strategies for them. descending to a ratio $(1+\alpha)(k+1)$ which depends on a parameter $\alpha \leq 1$.

We complete the article overview by saying that in Section 2 we remind the definitions of the k-CTP and the competitive ratio. We also draw the initial topography. Eventually, in Section 5, we conclude and give ideas about further work.

2 Preliminaries: notation and initial topography

We introduce the notation and remind the definitions of the k-CTP and the competitive ratio. Then, we show the topography constructed for strategies existing in the literature.

2.1 Notation

From now on, we focus on the k-CTP. A traveller traverses an undirected weighted graph $G = (V, E, \omega)$, $n = |V|$ and $m = |E|$. He starts his walk at source $s \in V$. His objective is to reach target $t \in V$ with a minimum cost (also called distance), which is the sum of the weights of edges traversed. At most k edges of G are *blocked*, which means that when the traveller reaches an endpoint of one of these edges, he discovers that he cannot pass through it. We note $E_* \subsetneq E$ the set of blocked edges, $|E_*| \leq k$. A pair (G, E_*) is called a *road map*. An *instance* is composed of a road map, nodes s and t and parameter k. We suppose that the instances treated are satisfiable: there is always a (s,t)-path in the graph $G \backslash E_*$, which represents graph G with all obstructed edges E_* withdrawn. If P signifies a path, $\omega(P)$ is the cost of this path. We note $\omega_{\min}$ the cost of the shortest (s,t)-path in graph G. For any subset of blocked edges already discovered $E' \subsetneq E_*$, $\omega_{\min}(E')$ is the cost of the shortest (s,t)-path in graph $G \backslash E'$. Value $\omega_{\text{opt}} = \omega_{\min}(E_*)$ is the *optimal offline cost* for map (G, E_*) and the corresponding path is called the *optimal offline path*. Figure 1 illustrates a walk of the traveller on a road map with two blockages. We remind the

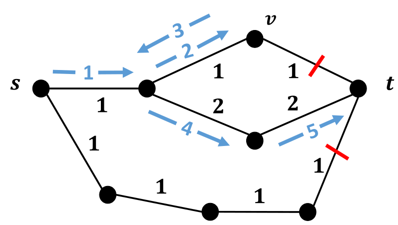


Figure 1: The walk of the traveller (in blue) on a graph with $k = 2$ blocked edges (red cuts). At node v, the traveller discovers the blocked edge (v, t).

definition of the competitive ratio introduced in [5] and specified for the k-CTP in [6]. Let $\omega_A(E_*)$ be the distance traversed by the traveller guided by a given strategy A on graph G from source s to target t with blocked edges E_*. Strategy A is c_A-competitive if, for any road map (G, E_*):

$$\omega_A(E_*) \leq c_A \omega_{\min}(E_*) + \eta,$$

where η is constant. For randomized strategies, this formula becomes:

$$\mathbb{E}[\omega_A(E_*)] \leq c_A \omega_{\min}(E_*) + \eta.$$

To assess strategies more precisely, we propose to specify these definitions for a family of instances. For example, let $\mathcal{G}$ denote the family of instances with graph G composed exclusively of node-disjoint (s,t)-paths. We say that strategy A is $c_{A,\mathcal{G}}$-competitive on $\mathcal{G}$ if and only if (iff), for any road map (G, E_*) with $G \in \mathcal{G}$, $\omega_A(E_*) \leq c_{A,\mathcal{G}} \omega_{\min}(E_*) + \eta$. As evoked in Section 1, the randomized REPOSITION from [3] is $(k+1)$-competitive for family of instances $\mathcal{G}$.

In this paper, we focus on *polynomial-time* strategies, executed in time $\mathcal{O}(p(n, m, k))$, where p is a polynomial, in order to favor a short execution time. One might also consider FPT strategies, *i.e.* strategies that are executed in time $\mathcal{O}(f(k)p(n, m))$, where p is a polynomial and f an arbitrary function [10]. However, no FPT strategy was proposed in the literature yet.

2.2 Topography of the competitiveness of existing strategies

We draw the topography of existing strategies on the set of all k-CTP instances. As explained, on the set of instances $\mathcal{G}$ composed of node-disjoint (s,t)-paths graphs only, there is a strategy A with $c_{A,\mathcal{G}} = k+1$. Otherwise, all we know is that, with REPOSITION and COMPARISON, the traveller does not traverse a distance greater than $(2k+1)\,\omega_{\text{opt}}$. We do not know whether there are particular families of instances on which these strategies (or others) achieve a ratio lower than $2k+1$. We represent this analysis graphically in Figure 2. Instances composed of node-disjoint (s,t)-paths graphs are represented in green, the other instances are in white. One who wants to

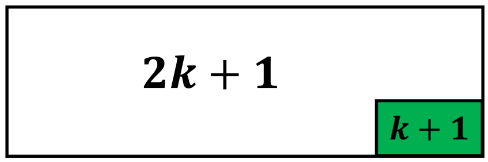


Figure 2: Topography of the competitiveness on the k-CTP before this study started

solve the k-CTP on an instance in the white zone would only know that the traveller cannot traverse the graph with a distance greater than $(2k+1)\,\omega_{\text{opt}}$ using existing strategies. The inconvenience comes from the definition of the competitive ratio, which evaluates the quality of a strategy on the worst instance. For example, the competitive ratio of GREEDY strategy [13] is $\mathcal{O}\left(2^k\right)$, but it becomes $\mathcal{O}\left(1\right)$ for a certain class of grids. From a practical point of view, the competitive analysis does not give enough information on the quality of a strategy on a given instance.

Our goal is to determine instance families which can be solved by strategies A with "local" $c_A < 2k+1$. We thus propose in Section 3 a criterion such that there is a strategy A solving the instances which satisfy it with "local" $c_A < 2k+1$.

An example of relevant criterion is the one which characterizes instances composed of node-disjoint (s,t)-paths graphs. This criterion can be verified in time $\mathcal{O}\left(n\right)$ and the competitive ratio of the randomized REPOSITION proposed in [3] on graphs which satisfy it is $k+1$. However, this family of graphs is a very specific case and does not have many practical applications. On the contrary, the pseudo-grid criterion proposed in [13] cannot be verified in polynomial time and, moreover, GREEDY strategy solves it with exponential competitive ratio α^k, where $\alpha \geq 1$ depends on the instance.

We will complete the topography by identifying sets of instances for which we propose strategies with a competitive ratio less than $2k+1$. In the next section, we devise a deterministic strategy which achieves a ratio lower than $2k+1$ for graphs which satisfy a criterion $C\left(\gamma\right)$.

3 Completing the topography with PIVOT-REPOSITION

We propose a deterministic strategy called PIVOT-REPOSITION. It is based upon the selection of a pivot v^*, a set $\mathcal{P}_{s,v^*}$ of edge-disjoint (s,v^*)-paths and a set $\mathcal{P}_{v^*,t}$ of edge-disjoint (v^*,t)-paths. Once we obtain them, this strategy consists in executing REPOSITION strategy on paths $\mathcal{P}_{s,v^*}$ and, if the traveller reaches v^*, executing REPOSITION on $\mathcal{P}_{v^*,t}$. We describe our algorithm (Section 3.1), the selection of the triplet $(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t})$ it uses (Section 3.2) and the criterion $C\left(\gamma\right)$ (Section 3.3). PIVOT-REPOSITION reaches the $(2-\gamma)\,k+1$ competitive ratio on instances satisfying $C\left(\gamma\right)$.

3.1 Description of the PIVOT-REPOSITION strategy

PIVOT-REPOSITION uses a preprocessing algorithm to pick out node v^*, called *pivot*. Given node v^*, it also selects two path sets: $\mathcal{P}_{s,v^*}$ which contain edge-disjoint (s, v^*)-paths and $\mathcal{P}_{v^*,t}$ which contain edge-disjoint (v^*, t)-paths. These elements are chosen in order to minimize the distance traversed by the traveller with this strategy (Section 3.2). Figure 3 gives an example of pivot v^* and edge-disjoint paths $\mathcal{P}_{s,v^*}$ and $\mathcal{P}_{v^*,t}$ on a graph G.

1: **Input:** graph G, source s, target t, positive integer k
2: $(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t}) \leftarrow$ PREPROCESSING(G);
3: $l_1 \leftarrow |\mathcal{P}_{s,v^*}|$; $l_2 \leftarrow |\mathcal{P}_{v^*,t}|$; $i \leftarrow 1$;
4: **while** *the traveller does not reach* v^* **and** $i \leq l_1$ **do**
5:     the traveller traverses the i^{th} shortest path of $\mathcal{P}_{s,v^*}$;
6:     **if** *the traveller is blocked* **then** he returns to s; $i \leftarrow i + 1$ **endif**
end
7: **if** *the traveller is standing at node* v^* **then**
8:     $j \leftarrow 1$;
9:     **while** *the traveller does not reach* t **and** $j \leq l_2$ **do**
10:         the traveller traverses the j^{th} shortest path of $\mathcal{P}_{v^*,t}$;
11:         **if** *the traveller is blocked* **then** he returns to v^*; $j \leftarrow j + 1$ **endif**
    end
12: **endif**
13: $E' \leftarrow$ the set of blocked edges discovered by the traveller;
14: **if** *the traveller did not reach* t **then** PIVOT-REPOSITION$(G \backslash E')$ **endif**

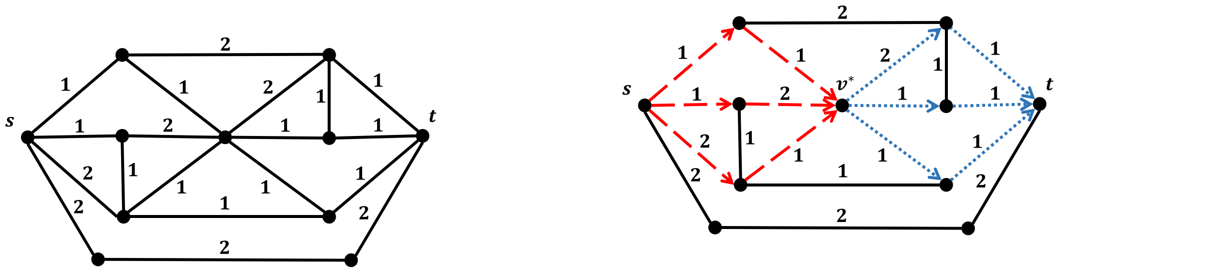
Algorithm 1: The PIVOT-REPOSITION strategy

PIVOT-REPOSITION is given in Algorithm 1. We specify in Section 3.2 the pivot-based instance preprocessing MINCOST, which is the algorithm to pick out node v^* and sets $\mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t}$, run in line 2. Next, the traveller executes REPOSITION on paths in $\mathcal{P}_{s,v^*}$ (lines 3–6) until he arrives at v^*. He traverses the shortest path in $\mathcal{P}_{s,v^*}$ and, if he is blocked, comes back to s and traverses the second shortest path in $\mathcal{P}_{s,v^*}$, etc. If the traveller does not reach v^*, which means that all paths of $\mathcal{P}_{s,v^*}$ are blocked, he restarts the process by selecting another triplet, taking into account the edges discovered during his unsuccessful attempts (lines 13–14). Otherwise, once he reaches v^*, REPOSITION is executed on $\mathcal{P}_{v^*,t}$ (lines 8–12). Similarly, if he does not reach t starting from the pivot (all paths in $\mathcal{P}_{v^*,t}$ are blocked), he returns to s and selects another triplet (lines 13–14).

3.2 Pivot-based preprocessing of an instance

We propose a preprocessing algorithm (executed in line 2 in Algorithm 1) called MINCOST. Appendix A reminds methods which produce the largest set of edge-disjoint paths between two arbitrary nodes or the set of edge-disjoint paths which minimizes the sum of the costs of these paths. These methods are used by MINCOST to identify the triplet $(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t})$ which reduces the competitive ratio of PIVOT-REPOSITION.

We introduce the notation needed to characterize any node v in graph G. For any set $\mathcal{P}_{s,v}$ composed of l_1 edge-disjoint (s, v)-paths ordered according to their cost, we note $P_{s,v}^{(i)}$ the i^{th} shortest path in $\mathcal{P}_{s,v}$ and $\omega_{s,v}^{(i)}$ is its cost. Let $\overline{\omega_{s,v}}$ denote the average cost of paths in set $\mathcal{P}_{s,v}$. For any set $\mathcal{P}_{v,t}$, composed of l_2 edge-disjoint (v, t)-paths, we use $P_{v,t}^{(i)}$, $\omega_{v,t}^{(i)}$ and $\overline{\omega_{v,t}}$, respectively.



(a) Graph G

(b) Pivot v^* and sets $\mathcal{P}_{s,v^*}$ (dashed red paths) and $\mathcal{P}_{v^*,t}$ (dotted blue paths).

Figure 3: Pivot v^* and two sets of edge-disjoint (s, v^*)-paths $\mathcal{P}_{s,v^*}$ and edge-disjoint (v^*, t)-paths $\mathcal{P}_{v^*,t}$ in graph G

At the beginning, we assume not only that the triplet $(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t})$ is known but also that the competitive ratio of PIVOT-REPOSITION is less or equal to $2k+1$. We establish an upper bound of the competitive ratio of PIVOT-REPOSITION which is potentially lower than $2k+1$ and which is a function of the mean costs $\overline{\omega_{s,v^*}}$ and $\overline{\omega_{v^*,t}}$. This bound will help us to select the triplet $(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t})$. We distinguish three scenarii of the PIVOT-REPOSITION execution on graph G.

- Scenario 1: **all paths in $\mathcal{P}_{s,v^*}$ are blocked**. This may only happen when $l_1 \leq k$. The traveller thus traverses each path in $\mathcal{P}_{s,v^*}$ and comes back to s after discovering a blockage (lines 3–6 of Algorithm 1). The distance traversed is less than $2l_1\overline{\omega_{s,v^*}}$. Observe that $2l_1\frac{\overline{\omega_{s,v^*}}}{\omega_{\text{opt}}} \leq 2l_1\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}}$ as the cost of the shortest path in G, $\omega_{\min}$, cannot be worse than the optimal offline cost ω_{opt}. Then, the strategy is applied on graph G deprived of the l_1 blockages discovered (line 14). This implies that the distance traversed by the traveller on this remaining graph is at worst $2(k-l_1)+1$ times greater than ω_{opt}. Function H_1 gives an upper bound of the competitive ratio when $\mathcal{P}_{s,v^*}$ is fully blocked:

$$H_1(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t}) = 2l_1\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}} + 2(k-l_1) + 1. \tag{1}$$

- Scenario 2: **the traveller reaches v^* but all paths in $\mathcal{P}_{v^*,t}$ are blocked**. This may happen when $l_2 \leq k$. Let us assume that the traveller reached v^* passing through a certain path $P^{(r)}_{s,v^*}$. In other words, r, $1 \leq r \leq l_1$ is the index of the first path in $\mathcal{P}_{s,v^*}$ which is open (it is the shortest available). The traveller will also use it to go back to s after detecting that all paths in $\mathcal{P}_{v^*,t}$ are obstructed. The total distance is thus $2\sum_{i=1}^{r}\omega^{(i)}_{s,v^*} + 2l_2\overline{\omega_{v^*,t}}$ which can be bounded from above by $2r\overline{\omega_{s,v^*}} + 2l_2\overline{\omega_{v^*,t}}$, as $\sum_{i=1}^{r}\omega^{(i)}_{s,v^*} \leq r\overline{\omega_{s,v^*}}$. At this moment, $r-1+l_2$ blockages have already been discovered and PIVOT-REPOSITION is applied to graph G deprived of them. Function H_2 is an upper bound of the competitive ratio in this case:

$$H_2(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t}) = \max_{1\leq r\leq l_1}\left(2r\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}} + 2l_2\frac{\overline{\omega_{v^*,t}}}{\omega_{\min}} + 2(k-(r-1)-l_2) + 1\right). \tag{2}$$

We can compute function H_2 in time $\mathcal{O}(1)$ instead of $\mathcal{O}(k)$. The method to obtain this simplification is detailed in Appendix B.

$$H_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right) = \max\left\{h_{21}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right), h_{22}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)\right\}, \text{ where} \tag{3}$$

$$h_{21}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right) = 2\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}} + 2l_2\frac{\overline{\omega_{v^*,t}}}{\omega_{\min}} + 2\left(k-l_2\right)+1, \tag{4}$$

$$h_{22}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right) = 2l_1\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}} + 2l_2\frac{\overline{\omega_{v^*,t}}}{\omega_{\min}} + 2\left(k-l_2-l_1+1\right)+1. \tag{5}$$

- Scenario 3: **the traveller reaches t passing through v^*.** In other words, at least one path in $\mathcal{P}_{s,v^*}$ and one path in $\mathcal{P}_{v^*,t}$ are open. Let r_1 be the index of the shortest path in $\mathcal{P}_{s,v^*}$ which is open. The index of the shortest open path in $\mathcal{P}_{v^*,t}$ is r_2. The traveller is blocked r_1-1 times before reaching v^* and then r_2-1 times after v^*. As the number of blockages is bounded by k, this implies $r_1+r_2-2 \leq k$. The worst situation occurs when the traveller is confronted with all k blockages. The distance traversed is at most $2\sum_{i=1}^{r_1-1}\omega_{s,v^*}^{(i)} + \omega_{s,v^*}^{(r_1)} + 2\sum_{i=1}^{r_2-1}\omega_{v^*,t}^{(i)} + \omega_{v^*,t}^{(r_2)}$ which is less than $(2r_1-1)\,\overline{\omega_{s,v^*}} + (2r_2-1)\,\overline{\omega_{v^*,t}}$. Function H_3 is an upper bound of the competitive ratio in this case:

$$H_3\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right) = \max_{\substack{1\leq r_1\leq l_1,\ 1\leq r_2\leq l_2\\ r_1+r_2-2=\min\{k,l_1+l_2-2\}}} \left(2r_1-1\right)\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}} + \left(2r_2-1\right)\frac{\overline{\omega_{v^*,t}}}{\omega_{\min}}.$$

  Value $H_3\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)$ is the maximum of a linear function with respect to variables ω_{s,v^*} and $\omega_{v^*,t}$ over a convex admissible set of dimension 1. Therefore, the maximum is located at one of the extremity of this set, which corresponds at most to $\{r_1=1, r_2=k+1\}$ or $\{r_1=k+1, r_2=1\}$. Function H_3 can thus be rewritten:

$$H_3\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right) = \max\left\{\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}} + (2k+1)\frac{\overline{\omega_{v^*,t}}}{\omega_{\min}}, (2k+1)\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}} + \frac{\overline{\omega_{v^*,t}}}{\omega_{\min}}\right\}. \tag{6}$$

The three scenarii cover all the possible executions of PIVOT-REPOSITION. Functions H_1, H_2, H_3 are upper bounds of the competitive ratio of this strategy under Scenario 1, 2, 3 respectively. Minimization of the competitive ratio of PIVOT-REPOSITION necessitates the minimization of the bound H computed in the worst case of H_1, H_2, and H_3 which depends on the triplet $(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t})$, as summarized in Table 1. Computing H is in $\mathcal{O}(1)$ as all H_i, $i=1,2,3$, expressed by Eqs. (1), (3) and (6) are in $\mathcal{O}(1)$.

Condition	$\boldsymbol{H\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)}$
$l_1>k$ and $l_2>k$	$H_3\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)$
$l_1\leq k$ and $l_2>k$	$\max_{i\in\{1,3\}} H_i\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)$
$l_1>k$ and $l_2\leq k$	$\max_{i\in\{2,3\}} H_i\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)$
$l_1\leq k$ and $l_2\leq k$	$\max_{i\in\{1,2,3\}} H_i\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)$

Table 1: Upper bound H in function of scenario-dependent H_i, $i=1,2,3$.

To insert pivot v^*, we propose a polynomial-time algorithm MINCOST which minimizes function H. Let us suppose that pivot v^* and values l_1,l_2 are known while sets $\mathcal{P}_{s,v^*}$ and $\mathcal{P}_{v^*,t}$ are to be determined. According to Eqs. (1) and (4)–(6), function H is minimized iff $\overline{\omega_{s,v^*}}$ and $\overline{\omega_{v^*,t}}$ are minimum. For a fixed l_1, the set $\mathcal{P}_{s,v^*}$ which minimizes the mean cost $\overline{\omega_{s,v^*}}$ represents the minimum cost l_1 edge-disjoint (s,v)-paths which are identified in polynomial time (Appendix A).

1: **Input:** graph G, source s, target t, positive integer k
2: **Output:** $(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t})$
3: $H_{\min} \leftarrow +\infty$; $v^* \leftarrow s$
4: **foreach** $v \in G$ **do**
5: $\quad M_{s,v} \leftarrow$ minimum between $k+1$ and the maximal number of edge-disjoint (s,v)-paths;
6: $\quad M_{v,t} \leftarrow$ minimum between $k+1$ and the maximal number of edge-disjoint (v,t)-paths;
7: $\quad$ **for** $l_1 = 1 \ldots M_{s,v}$ **do**
8: $\quad\quad$ **for** $l_2 = 1 \ldots M_{v,t}$ **do**
9: $\quad\quad\quad \mathcal{P}_{s,v}^{(l_1)} \leftarrow$ minimum total cost l_1 edge-disjoint (s,v)-paths;
10: $\quad\quad\quad \mathcal{P}_{v,t}^{(l_2)} \leftarrow$ minimum total cost l_2 edge-disjoint (v,t)-paths;
11: $\quad\quad\quad$ **if** $H\left(v, \mathcal{P}_{s,v}^{(l_1)}, \mathcal{P}_{v,t}^{(l_2)}\right) < H_{\min}$ **then** $v^* \leftarrow v$; $H_{\min} \leftarrow H(v, \mathcal{P}_1, \mathcal{P}_2)$ **endif**
12: $\quad\quad$ **endfor**
13: $\quad$ **endfor**
14: **endforeach**

Algorithm 2: The pivot inserting MINCOST algorithm used in Algorithm 1

This reasoning also holds with l_2 and set $\mathcal{P}_{v^*,t}$. This is why MINCOST first enumerates all nodes $v \in G$ and values l_1, l_2, then determines triplet $(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t})$ with $|\mathcal{P}_{s,v}| = l_1$ and $|\mathcal{P}_{v,t}| = l_2$ which minimizes $\overline{\omega_{s,v}}$ and $\overline{\omega_{v,t}}$. The algorithm only considers that l_1 and l_2 cannot exceed $k+1$. Indeed, in a set of $k+2$ edge-disjoint (s,v)-paths $\mathcal{P}_{s,v}$ (or (v,t)-paths $\mathcal{P}_{v,t}$), the longest path, indexed by $k+2$, is never traversed.

The MINCOST algorithm (Algorithm 2), identifies, for any node v of graph G and any l_1, l_2, $1 \leq l_1, l_2 \leq k+1$, the sets of edge-disjoint (s,v)-paths of size l_1 and (v,t)-paths of size l_2 which minimize $\overline{\omega_{s,v}}$ and $\overline{\omega_{v,t}}$ respectively (lines 9 and 10). On each iteration, it compares value $H(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t})$ with the minimum value of $H_{\min}$ determined until now (line 11). Lines 5 and 6 determine admissible values for l_1 and l_2 with the Ford-Fulkerson's algorithm to obtain the maximal number of edge-disjoint (s,v)-paths and (v,t)-paths (Appendix A) which is of $\mathcal{O}(nm)$. Identifying sets $\mathcal{P}_{s,v}^{(l_1)}$ in and $\mathcal{P}_{v,t}^{(l_2)}$ (lines 9–10) is done in $\mathcal{O}(2nk(m + n\log n))$. The complexity of MINCOST is of $\mathcal{O}\left(n\left(2nm + k^2(2nk(m + n\log n))\right)\right) = \mathcal{O}\left(n^2k^3(m + n\log n)\right)$.

3.3 The competitiveness of the PIVOT-REPOSITION strategy

We identify a family of graphs for which the competitive ratio of PIVOT-REPOSITION drops below the $2k+1$ bound. An instance is in family $C(\gamma)$, where $0 \leq \gamma \leq 1$ iff there is a triplet $(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t})$ such that

$$\gamma \leq \frac{2k+1-H(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t})}{k}. \tag{7}$$

We are able to determine in polynomial time for which values of γ an instance belongs to $C(\gamma)$. Any instance is in $C(0)$. Given an instance, there is always a maximum value $\gamma_{\max} \geq 0$ such that, for any $\gamma \leq \gamma_{\max}$, the instance is in $C(\gamma)$. The MINCOST algorithm produces an optimal triplet $(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t})$. Then, thanks to Eq. (7), we find $\gamma_{\max} = \frac{1}{k}(2k+1-H(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t}))$. Figure 7a in Appendix C shows an example of graph G with $\gamma_{\max} = \frac{2}{1+\beta}$, where $\beta \geq 1$ is the weight of edges in G. Observe that $\gamma_{\max} \leq 1$. Indeed, for any triplet $(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t})$, $\omega_{s,v} + \omega_{v,t} \geq \omega_{\min}$ so, according to Eq. (6), $H_3(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t}) \geq k+1$ and, consequently $H(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t}) \geq k+1$. The local competitive ratio of PIVOT-REPOSITION is given by the theorem below. To respect the page limit, all proofs of theorems in this section are put in Appendix C.

Theorem 1 $c_{\text{PIVOT-REPOSITION},\ C(\gamma)} = (2-\gamma)\,k + 1.$

We can identify graphs, characterized by criterion $C(\gamma)$, for which the competitive ratio of PIVOT-REPOSITION is strictly smaller than $2k+1$. We prove that PIVOT-REPOSITION is the optimal deterministic strategy for the instances satisfying $C(\gamma)$.

Theorem 2 *There is no deterministic strategy which a competitive ratio lower than $(2-\gamma)\,k+1$ for the family of instances satisfying $C(\gamma)$.*

The global competitive ratio of PIVOT-REPOSITION with MINCOST inserting a pivot is $2k+1$.

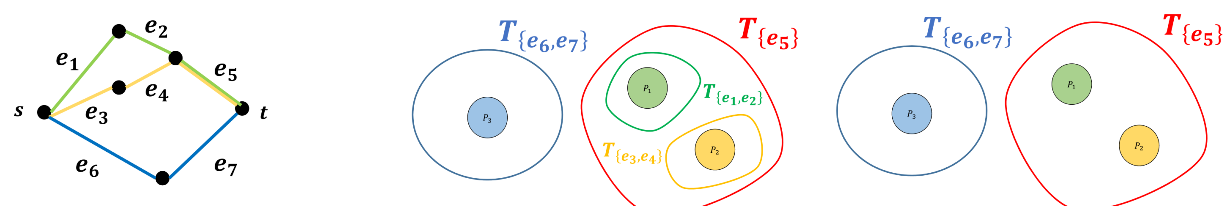
Theorem 3 $c_{\text{PIVOT-REPOSITION}} = 2k + 1.$

4 Competitive randomized strategies for families of instances

We complete the topography by extending the set of instances for which the competitive ratio is less than $2k+1$, despite the fact that they do not satisfy $C(\gamma)$ with $\gamma > 0$.

We define the *hypergraph representation* of a graph G with $m = |E|$. Let $H_G = \{T_{e_1}, \ldots, T_{e_m}\}$ be the associated hypergraph such that an (s,v)-path $P_i \in T_{e_j}$ iff path P_i contains edge e_j. In other words, hyperedge $T_e \in H_G$ is composed of paths which share edge e.

A simple hypergraph is a hypergraph such that no edge contains another one [4]. Put differently, for hyperedges T_i, T_j, $T_i \subset T_j \Rightarrow i = j$. We note H_G^+ the simple hypergraph representing graph G. It is obtained by removing hyperedges T_{e_i} from H_G such that there exists $j \neq i$, $T_{e_i} \subsetneq T_{e_j}$. If there are e_i, e_j such that $T_{e_i} = T_{e_j}$, we merge the two edges into a single one $T_{\{e_1,e_2\}}$. As m is the number of hyperedges of H_G, $m^+ = \left|H_G^+\right|$ is the number of hyperedges in hypergraph H_G^+, $H_G^+ = \left\{T_{E_1}, \ldots, T_{E_{m^+}}\right\}$. Figure 4 shows a graph G with its hypergraphs H_G and H_G^+.



(a) Graph G with 3 (s,t)-paths (green, yellow, and blue)

(b) Hypergraph H_G

(c) Simple hypergraph H_G^+

Figure 4: An illustration of the hypergraph representation: G together with H_G and H_G^+

The hypergraph representation brings out certain graphs properties which can be successfully exploited when designing randomized strategies. In this way, we identify two families of graphs:

- Graph G is in family $F_{\text{d}}(q)$ iff the hypergraph H_G^+ is composed of at least q connected components. Formally, set $\left\{T_{E_1}, \ldots, T_{E_{m^+}}\right\}$ can be partitioned in at least q groups which do not share any path. We say that graph G is *hyperedge-disjoint* iff it belongs to $F_{\text{d}}(q)$ with $q \geq k+1$. For instance, graph G represented in Figure 4a belongs to $F_{\text{d}}(2)$. We propose a randomized strategy dedicated to treat the family $F_{\text{d}}(k+1)$ in Appendix D.

- Graph G is in family F_{rp} iff hypergraph H_G is a *recursive partition*, *i.e.* for any hyperedges T_{e_i} and T_{e_j}, $T_{e_i} \cap T_{e_j} \neq \emptyset \Rightarrow T_{e_i} \subseteq T_{e_j} \vee T_{e_j} \subseteq T_{e_i}$. We design a competitive randomized strategy for this family in Appendix E where its example is also given in Figure 10b.

5 Conclusion and further work

Conclusion We constructed strategies with a competitive ratio which improves the $2k+1$ bound for families of k-CTP instances. We introduced the topography representation which assesses the competitiveness of strategies for k-CTP instances. Strategies are studied in the literature with the competitive ratio. The topography, based on the local evaluation of the competitive ratio, allows us to improve the competitiveness on certain instances and to adapt strategies to instances treated. We proposed families of instances which can be recognized in polynomial time and designed, for each of them, a strategy with a competitive ratio smaller than $2k+1$.

We built the deterministic strategy PIVOT-REPOSITION executed in polynomial time. We introduced a family of instances, characterized by criterion $C(\gamma)$, $0 \leq \gamma \leq 1$, such that we can verify whether graph G with bound k belongs to $C(\gamma)$ in polynomial time. The competitive ratio of PIVOT-REPOSITION for instances satisfying $C(\gamma)$ is $(2-\gamma)k+1$ and there is no deterministic strategy more competitive. Moreover, PIVOT-REPOSITION remains optimal on all k-CTP instances with a competitive ratio $2k+1$ which cannot be improved by deterministic strategy.

We proposed the representation of k-CTP instances as hypergraphs. This new vision allowed us to identify two categories of instances: hyperedge-disjoint and recursive-partition. We proved that, for hyperedge-disjoint graphs, there is a randomized strategy with competitive ratio $(1+\alpha)(k+1)$, where $\alpha \geq 0$ is a parameter of the graph. We also proposed the randomized strategy RP-REPOSITION which achieves a competitive ratio $(1+\alpha)(k+1)$ on recursive-partition graphs where α is the ratio between the cost of the longest (s,t)-path and the cost of the shortest (s,t)-path in graph G. These two families of instances are added to the topography whose final version is presented in Figure 5.

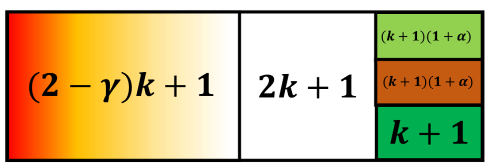


Figure 5: Topography of the competitiveness for the k-CTP after completing our study

Further work This study opens many research directions. First, the procedure which consists in adapting strategies to a family of instance recognizable in polynomial time, is novel and constitutes another way to assess the quality of a strategy. Moreover, this evaluation is more relevant from the practical point of view. Indeed, one can know which strategy is efficient for a given instance with the topography, whereas the competitive ratio judges a strategy on the worst instance possible.

PIVOT-REPOSITION, based on the selection of a pivot v^*, could be improved by picking out a set of pivots instead. The advantage is that it can cover a wider part of a graph, especially in the case of a large urban network. Although this direction seems very promising, identifying a criterion on which a strategy based on PIVOT-REPOSITION could be more competitive than $(2-\gamma)k+1$ is far from being obvious. Another possibility is to focus on randomized strategies to complete our topography. For some instances, the randomization is appropriate to make the competitive ratio decrease. For example, random draws could be integrated into PIVOT-REPOSITION.

Eventually, the hypergraph representation may be considered as a model for design, dimensioning, and reliability analysis of telecommunication or transportation networks.

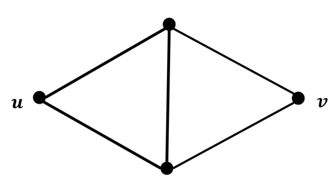
Appendix A Network flow algorithms as a tool

To explain how pivot v^* and paths in $\mathcal{P}_{s,v^*}$ and $\mathcal{P}_{v^*,t}$ are chosen, we remind the method of establishing whether there exists (or not) l edge-disjoint (u,v)-paths between nodes u and v of G, where $l \geq 1$. Based on the literature, we explain the manner to pick out l edge-disjoint (u,v)-paths which minimizes their total cost (sum of the cost of each path). The problem of finding the largest number of edge-disjoint paths between two nodes is formalized below.

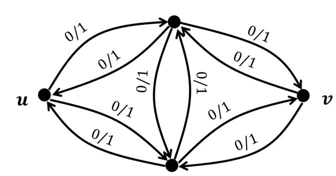
Problem 1 (Edge-disjoint paths problem)
Input: Graph G, nodes $u, v \in V$
Objective: Determine the maximum number $L_{u,v}$ of edge-disjoint (u,v)-paths.

This problem can be reduced to the MAXIMUM FLOW problem [8]. The reduction is detailed below and illustrated in Figure 6.

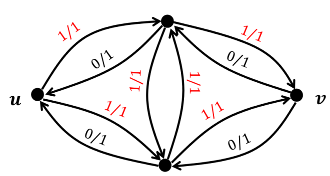
- Step 1: Transform graph G into a directed graph $\overrightarrow{G}$ by replacing an undirected edge (i,j) by two arcs (i,j) and (j,i) (Figure 6a).
- Step 2: Assign the unitary capacity to each arc in graph $\overrightarrow{G}$. Graph $\overrightarrow{G}$ becomes a network with the source u and the sink v (Figure 6b).
- Step 3: Obtain the integer-valued maximum flow $f_{\max}$ with Ford-Fulkerson's algorithm [7] (Figure 6c). Let $f_{\max}(i,j)$ denote the value of flow for arc (i,j) for the obtained MAXIMUM FLOW solution. The set of $L_{u,v}$ edge-disjoint (u,v)-paths is identified by picking out edges (i,j) of G such that flow values in symmetric arcs (i,j) and (j,i) are different, *i.e.* $|f_{\max}(i,j) - f_{\max}(j,i)| = 1$ (Figure 6d).



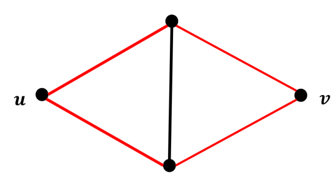
(a) Objective: identifying the largest number of edge-disjoint (u,v)-paths in graph G.



(b) After steps 1 and 2: network $\overrightarrow{G}$ with capacities equal to one.



(c) After step 3: we obtain the maximum flow which is 2.



(d) We deduce the largest set of edge-disjoint (u,v)-paths.

Figure 6: An illustration of the process to identify l edge-disjoint (u,v)-paths.

Menger's theorem [9] states that the largest number of edge-disjoint (u,v)-paths is equal to the minimal number of edges that separate u and v, which corresponds to the minimal cut of $\overrightarrow{G}$. As a consequence of the Max Flow-Min Cut theorem [7], the maximal flow is equal to $L_{u,v}$. The reduction and the identification of the maximal flow takes $\mathcal{O}(nm)$ which results from the analysis of Ford-Fulkerson's algorithm [1].

We define the problem to find the set of $l \leq L_{u,v}$ edge-disjoint (u,v)-paths of the minimum total cost.

Problem 2 (Minimum cost edge-disjoint (u,v)-paths)
Input: Graph G, nodes $u, v \in V$, positive integer $l \leq L_{u,v}$

Objective: Identify l edge-disjoint (u,v)-paths $P_1, P_2, \ldots, P_l$ which minimize $\sum_{i=1}^{l} \omega(P_i)$.

This problem can be reduced to the MINIMUM COST FLOW problem [1] whose definition is reminded below and which can be solved in polynomial time.

Problem 3 (Minimum cost flow problem [1])
Input: *Let* $\overrightarrow{G} = \left(V, \overrightarrow{E}, \omega\right)$ *be a weighted network with a source* u *and a sink* v. *A weight* $\omega_{i,j}$ *and a capacity* $c_{i,j}$, *which are both positive integers, are associated to arc* $(i,j) \in \overrightarrow{E}$. *For any node* $i \in V$, *value* $b(i)$, *which can be either positive or negative, specifies its supply/demand.*
Objective: *Minimize*

$$\sum_{(i,j) \in \overrightarrow{E}} \omega_{i,j} x_{i,j} \tag{8}$$

subject to

$$\forall i \in V, \sum_{j:(i,j) \in \overrightarrow{E}} x_{i,j} - \sum_{j:(j,i) \in \overrightarrow{E}} x_{j,i} = b(i) \quad \text{and} \quad \forall (i,j) \in \overrightarrow{E}, 0 \leq x_{i,j} \leq c_{i,j}.$$

As we want to identify the l minimum cost edge-disjoint (u,v)-paths in graph G, we transform it into an instance of the MINIMUM COST FLOW problem.

- Step 1: Transform G into network $\overrightarrow{G}$ with the source u, the sink v and the unitary capacity of all arcs. If (i,j) is an edge of graph G with weight $\omega_{i,j}$, weights of $(i,j) \in \overrightarrow{E}$ and $(j,i) \in \overrightarrow{E}$ are both equal to $\omega_{i,j}$.
- Step 2: Set $b(i) = 0$ for any node i, $i \neq u, v$, which forces the conservation of the flow. Set $b(u) = l$ and $b(v) = -l$ to specify the need to pick out exactly l edge-disjoint paths.
- Step 3: Solve the MINIMUM COST FLOW problem with the SUCCESSIVE SHORTEST PATH (SSP) algorithm [1]. SSP consists in solving $\mathcal{O}(nB)$ shortest path problems, where B is an upper bound of the largest supply, $B = l$ in our case. Edges in G, for which the identified minimum cost flow values of associated arcs (i,j) and (j,i) are different, represents the l edge-disjoint (u,v)-paths with minimal cost.

The identification of the l edge-disjoint paths from u to v which minimize the total cost is executed in time $\mathcal{O}(nl(m + n\log n))$ according to the complexity of SSP [1]. Table 2 summarizes the complexity of algorithms solving the two problems defined in this section.

Objective	Time
Identifying the largest set of edge-disjoint (u,v)-paths	$\mathcal{O}\left(nm\right)$
Identifying l edge-disjoint (u,v)-paths which minimize their total cost	$\mathcal{O}\left(nl\left(m+n\log n\right)\right)$

Table 2: Two flow problems defined and the complexity of algorithm solving them

Appendix B Function H_2 computed in time $\mathcal{O}\left(1\right)$

Theorem 4 $H_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)=\max\left\{h_{21}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right),h_{22}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)\right\}$ *where*

$$\begin{aligned}h_{21}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)&=2\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}}+2l_2\frac{\overline{\omega_{v^*,t}}}{\omega_{\min}}+2\left(k-l_2\right)+1,\\ h_{22}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)&=2l_1\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}}+2l_2\frac{\overline{\omega_{v^*,t}}}{\omega_{\min}}+2\left(k-l_2-l_1+1\right)+1.\end{aligned}$$

Proof. We set $h_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t},r\right)=2r\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}}+2\frac{\omega^{(r)}_{s,v^*}}{\omega_{\min}}+2l_2\frac{\overline{\omega_{v^*,t}}}{\omega_{\min}}+2\left(k-(r-1)-l_2\right)+1$, such that $H_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)=\max_{1\leq r\leq l_1}h_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t},r\right)$. We note $\Delta\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t},r\right)$ the finite difference of function h_2, put formally,

$$\Delta\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t},r\right)=h_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t},r+1\right)-h_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t},r\right).$$

One can check that $\Delta\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t},r\right)=2\left(\frac{\overline{\omega_{s,v^*}}}{\omega_{\min}}-1\right)$ which is constant. As a consequence, the maximum of h_2 over $1\leq r\leq l_2$ is located at the extremity of this interval:

$$H_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)=\max\left\{h_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t},1\right),h_2\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t},l_1\right)\right\}.$$

Referring to Eq. (2), we obtain functions $h_{21}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)$ and $h_{22}\left(v^*,\mathcal{P}_{s,v^*},\mathcal{P}_{v^*,t}\right)$. ■

Appendix C Proofs of theorems from Section 3.3

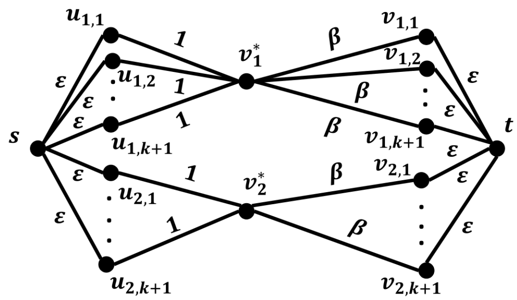
Theorem 1 $c_{\text{PIVOT-REPOSITION},\ C(\gamma)}=\left(2-\gamma\right)k+1$.

Proof. For any instance from $C\left(\gamma\right)$, there is a triplet $\left(v,\mathcal{P}_{s,v},\mathcal{P}_{v,t}\right)$ satisfying inequality (7) which means that $H\left(v,\mathcal{P}_{s,v},\mathcal{P}_{v,t}\right)\leq\left(2-\gamma\right)k+1$. The fact that the competitive ratio of PIVOT-REPOSITION is less than $H\left(v,\mathcal{P}_{s,v},\mathcal{P}_{v,t}\right)$ finishes the proof. ■

Theorem 2 *There is no deterministic strategy which a competitive ratio lower than* $\left(2-\gamma\right)k+1$ *for the family of instances* $C\left(\gamma\right)$.

Proof. We focus on $\gamma>0$ as all instances of k-CTP are in $C\left(0\right)$. We take as an example the instance with graph G depicted in Figure 7a. Our goal is to prove not only that (G,s,t,k) is in $C\left(\gamma\right)$ but also that there is no deterministic strategy which achieves a ratio lower than $\left(2-\gamma\right)k+1$ on this instance with at most k blocked edges.

Graph G is composed of $k+1$ edge-disjoint (s,v_1^*)-paths of cost $1+\varepsilon$, $k+1$ edge-disjoint (s,v_2^*)-paths of cost $1+\varepsilon$, $k+1$ edge-disjoint (v_1^*,t)-paths of cost $\beta+\varepsilon$ and $k+1$ edge-disjoint (v_2^*,t)-paths of cost $\beta+\varepsilon$. Values β and ε are two constants such that $\beta\geq1$ and $\varepsilon\ll1$, so ε will be omitted in formulæ when possible. The cost of the shortest (s,t)-path in G is $\omega_{\min}=1+\beta$.



(a) Graph G

	$\left(v_1^*, \mathcal{P}_{s,v_1^*}, \mathcal{P}_{v_1^*,t}\right)$	$\left(v_2^*, \mathcal{P}_{s,v_2^*}, \mathcal{P}_{v_2^*,t}\right)$
$\lvert\mathcal{P}_{s,v_i^*}\rvert$	$k+1$	$k+1$
$\lvert\mathcal{P}_{v_i^*,t}\rvert$	$k+1$	$k+1$
$\mathcal{P}_{s,v_i^*}$	$\{s \to u_{1,j} \to v_1^* \mid 1 \le j \le k+1\}$	$\{s \to u_{2,j} \to v_2^* \mid 1 \le j \le k+1\}$
$\mathcal{P}_{v_i^*,t}$	$\{v_1^* \to v_{1,j} \to t \mid 1 \le j \le k+1\}$	$\{v_2^* \to v_{2,j} \to t \mid 1 \le j \le k+1\}$
H_3	$H_3\left(v_1^*, \mathcal{P}_{s,v_1^*}, \mathcal{P}_{v_1^*,t}\right) = H_3\left(v_2^*, \mathcal{P}_{s,v_2^*}, \mathcal{P}_{v_2^*,t}\right) = (2k\beta+1+\beta)(1+\beta)^{-1}$	

(b) Analysis of the two optimal triplets $\left(v_1^*, \mathcal{P}_{s,v_1^*}, \mathcal{P}_{v_1^*,t}\right)$ and $\left(v_2^*, \mathcal{P}_{s,v_2^*}, \mathcal{P}_{v_2^*,t}\right)$

Figure 7: Graph G and summary of its optimal triplets

There are two optimal triplets returned by MINCOST, one with pivot v_1^* and another with pivot v_2^* (Figure 7b). We fix our attention on the triplet made with pivot v_1^*, $\left(v_1^*, \mathcal{P}_{s,v_1^*}, \mathcal{P}_{v_1^*,t}\right)$ as our analysis can directly be extended on pivot v_2^* as G has a horizontal axis of symmetry. As $\left|\mathcal{P}_{s,v_1^*}\right| = \left|\mathcal{P}_{v_1^*,t}\right| = k+1$, $H\left(v_1^*, \mathcal{P}_{s,v_1^*}, \mathcal{P}_{v_1^*,t}\right) = H_3\left(v_1^*, \mathcal{P}_{s,v_1^*}, \mathcal{P}_{v_1^*,t}\right)$. We deduce the $\gamma_{\max}$ from the expression of $H_3\left(v_1^*, \mathcal{P}_{s,v_1^*}, \mathcal{P}_{v_1^*,t}\right)$ obtained from Eq. (6) in Figure 7b:

$$\gamma_{\max} = \frac{2k+1-H_3\left(v_1^*, \mathcal{P}_{s,v_1^*}, \mathcal{P}_{v_1^*,t}\right)}{k} = \frac{2}{1+\beta}.$$

If we fix $\beta = \frac{2}{\gamma} - 1$ (equivalent to the formula above), our instance is in $C(\gamma)$. For any $0 \le \gamma \le 1$, it is possible to make graph G be in $C(\gamma)$ by adapting value β. We prove now that no deterministic strategy achieves a ratio smaller than $(2-\gamma)$ on G with at most k blocked edges.

We define sets $V_{v_1^*,t} = \{v_{1,1}, v_{1,2}, \ldots, v_{1,k+1}\}$ and $V_{v_2^*,t} = \{v_{2,1}, v_{2,2}, \ldots, v_{2,k+1}\}$. The traveller necessarily visits nodes in $V_{v_1^*,t} \cup V_{v_2^*,t}$ before reaching target t. We call the *traveller history* the record of visited nodes in $V_{v_1^*,t} \cup V_{v_2^*,t}$. For example, the traveller history $[v_{1,3}, v_{2,1}, v_{1,2}]$ means that the traveller visited node $v_{1,3}$, next $v_{2,1}$ and then $v_{1,2}$ before reaching t. With at most k blockages, the worst case occurs when, for the k first visited nodes $v_{i,j}$, all edges $(v_{i,j}, t)$ are blocked. The minimum distance between two nodes which are both in $V_{v_1^*,t}$ (or $V_{v_2^*,t}$) is 2β. At the same time, the minimum distance between any node in $V_{v_1^*,t}$ and any node in $V_{v_2^*,t}$ is $2+2\beta$. The optimal deterministic strategy selects all the $k+1$ first visited nodes of the traveller history either exclusively in $V_{v_1^*,t}$ or exclusively in $V_{v_2^*,t}$. If the traveller history is $[v_{1,1}, v_{1,2}, \ldots, v_{1,k+1}]$ and that edges $(v_{1,1}, t), \ldots, (v_{1,k}, t)$ are blocked, the traveller traverses at least a distance $1 + (2k+1)\beta$. Any simple (s,t)-path containing v_2^* is open so the optimal offline cost is $1+\beta$. No deterministic strategy does better than $\frac{1+(2k+1)\beta}{1+\beta} = (2-\gamma)k+1$ on instances satisfying $C(\gamma)$. ■

Theorem 3 $c_{\text{PIVOT-REPOSITION}} = 2k+1$.

Proof. We use the induction on the number of blockages. We start with $k = 0$. Let v be a node on the shortest (s,t)-path of graph G. Let $\mathcal{P}_{s,v}$ contain exclusively the shortest (s,v)-path of cost $\omega_{\min}^{(s,v)}$ and $\mathcal{P}_{v,t}$ contain the shortest (v,t)-path of cost $\omega_{\min}^{(v,t)}$. So, $H(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t}) = H_3(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t}) = 1$. As $H(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t}) \leq H(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t})$ by definition of the pivot which is located at an optimal position, the traveller reaches t with distance smaller than $\omega_{\min}$. Consequently, the competitive ratio is $1 = 2k+1$.

The second step of the induction consists in assuming that, for any road map with $k' < k$ blocked edges, $c_{\text{PIVOT-REPOSITION}} = 2k' + 1$. Our goal is to prove that the ratio of PIVOT-REPOSITION on graph G with k blocked edges is $2k+1$. Let $(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t})$ be the triplet produced by MINCOST. By definition of H and according to the induction hypothesis, $H(v^*, \mathcal{P}_{s,v^*}, \mathcal{P}_{v^*,t})$ is an upper bound on its competitive ratio.

We prove that the competitive ratio of PIVOT-REPOSITION is lower than $2k+1$ for a specific triplet. Let v be a node on the shortest (s,t)-path. Let $\mathcal{P}_{s,v}$ contain exclusively the shortest (s,v)-path and $\mathcal{P}_{v,t}$ contain the shortest (v,t)-path. We have $\omega_{\min}^{(s,v)} + \omega_{\min}^{(v,t)} = \omega_{\min}$. Consequently, according to the expressions for H_1, H_2, and H_3, $H(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t}) \leq 2k+1$. As the ratio goes below $2k+1$ for triplet $(v, \mathcal{P}_{s,v}, \mathcal{P}_{v,t})$, it can thus be even less for the MINCOST-chosen triplet. As PIVOT-REPOSITION is exactly $2k+1$ is deterministic, its competitive ratio cannot be strictly inferior to $2k+1$ [12], so $c_{\text{PIVOT-REPOSITION}} = 2k+1$. ■

Appendix D A randomized strategy targeting family $F_{\text{d}}(k+1)$

Figure 8 illustrates a graph G from $F_{\text{d}}(k+1)$. Graphs G_j, $1 \leq j \leq k+1$, are the connected components of $G \setminus \{s,t\}$. All hyperedges of H_G^+ issued from G can be partitioned into $k+1$ sets, noted $\mathcal{T}_j$. We arrange these sets such that $\mathcal{T}_j$ contain (s,t)-paths which traverse graph G_j.

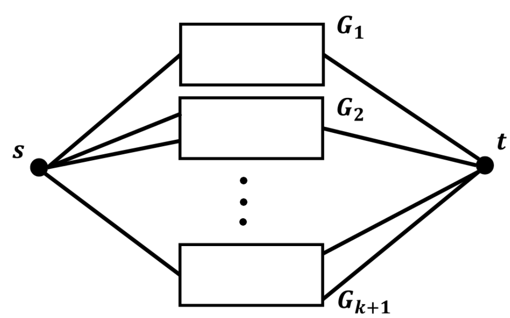


Figure 8: Graph G from $F_{\text{d}}(k+1)$, composed of disjoint subgraphs G_j, $1 \leq j \leq k+1$.

We identify the connected components G_j of graph $G \setminus \{s,t\}$ and determine a set $\mathcal{T}_j$ of hyperedges for each of them. At least one endpoint of edges e_i such that $T_{e_i} \in \mathcal{T}_j$ is a node of G_j. This polynomial-time method not only verifies whether graph G belongs to the family $F_{\text{d}}(k+1)$ but also identifies the connected components of H_G.

Theorem 5 *Let G be hyperedge-disjoint and $\omega_{j'}$ denote the weight of the shortest (s,t)-path of set $\mathcal{T}_j$. We assume for a certain α, $0 \leq \alpha < 1$ and any j, $\omega_{min} \leq \omega'_j \leq (1+\alpha)\,\omega_{min}$. There is a randomized strategy A such that $c_{A,F_d(k+1)} = (1+\alpha)(k+1)$.*

Proof. As $G \in F_{\text{d}}(k+1)$ and it contains at most k blocked edges, at least one set $\mathcal{T}_l$ is blockage-free: for any $T_{e_i} \in \mathcal{T}_l$, edge e_i is open. Our strategy starts by identifying the shortest (s,t)-path in each set $\mathcal{T}_l$. In this way, we obtain $k+1$ node-disjoint (s,v)-paths which are surely open. Next, the randomized REPOSITION strategy [3] is applied. The distance traversed by the traveller is bounded by $(k+1)(1+\alpha)\,\omega_{\min}$. As the optimal offline cost is greater or equal than $\omega_{\min}$, the competitiveness of this strategy is at worst $(1+\alpha)(k+1)$. ■

Appendix E A randomized strategy targeting family F_{rp}

We remind here that two hyperedges T_i and T_j, $i \neq j$, either are disjoint or one includes another. We list hyperedges of a recursive partition in the specific order to take into account the strict inclusion relation existing (or not) between them. First, we place all q disjoint hyperedges $T_1, T_2, \ldots, T_q$. Next, we put all r_1 hyperedges $T_{1,1}, \ldots, T_{1,r_1}$ which are strictly included in T_1. Hyperedges $T_{2,1}, \ldots, T_{2,r_2}$ strictly included in T_2 follow them, etc.

$$H_G = \left(T_1, T_2, \ldots, T_q, T_{1,1}, T_{1,2}, \ldots, T_{1,r_1}, T_{2,1}, \ldots, T_{q,r_q}, T_{1,1,1}, \ldots, T_{1,1,r_{11}}, \ldots\right), \tag{9}$$

where $T_{1,1}, T_{1,2}, \ldots, T_{1,r_1} \subsetneq T_1$, but also $T_{2,1}, \ldots, T_{2,r_2} \subsetneq T_2$ and $T_{1,1,1}, \ldots, T_{1,1,r_{11}} \subsetneq T_{1,1}$, etc.

List H_G may be seen as an arborescence, as illustrated in Figure 9. As $T_{1,1} \subseteq T_1$, set T_1 is the predecessor of $T_{1,1}$.

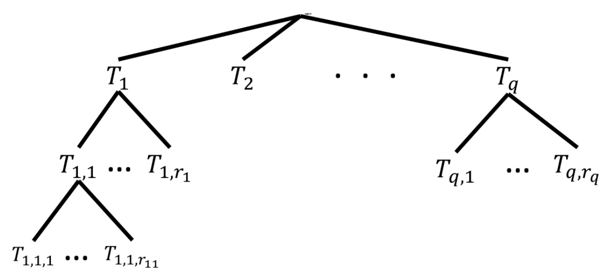
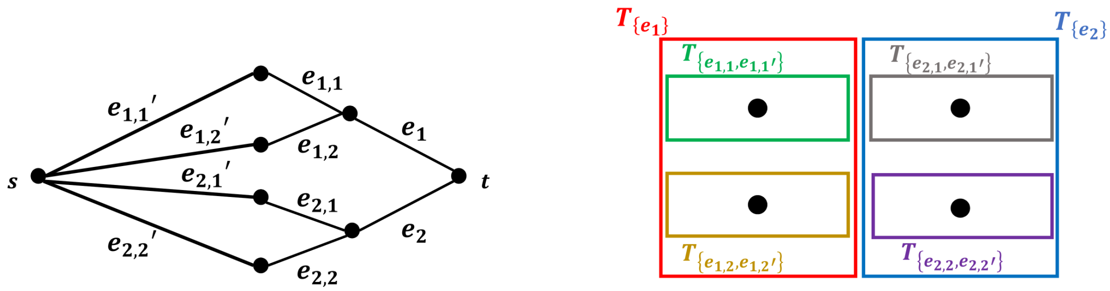


Figure 9: Hypergraph H_G, as an arborescence of its hyperedges

It is possible to verify whether a graph G is in family F_{rp} and to determine the hypergraph H_G in polynomial time if it is the case. Indeed, the number of (s,t)-paths in recursive-partition graphs is polynomial. The verification method is given below.

- If G is composed of a single (s,t)-path, return a positive answer which ends the verification. Otherwise, identify the connected components G_j in graph $G \setminus \{s,t\}$.
- For all G_j, check whether there is an edge e such that any (s,t)-path in $G_j \cup \{s,t\}$ belongs to hyperedge T_e. This is done by enumerating all (s,t)-paths in $G_j \cup \{s,t\}$ and verifying whether an edge e is in all of them.
- If it is the case, apply this method recursively to graphs $G_j \cup \{s,t\}$, otherwise return a negative answer which ends the verification.
- Return a positive answer after all (s,t)-paths in G have been visited and if no negative answer was sent until now.

We design a randomized strategy A for graphs G in family F_{rp} with $c_{A,F_{\mathrm{rp}}} = (k+1)(1+\alpha)$, where $\frac{\omega_i}{\omega_j} \leq 1+\alpha$ for any (s,t)-paths P_i, P_j of cost ω_i and ω_j, respectively. This generalizes the result from [6] obtained for apex trees with equal costs ($\alpha = 0$). Indeed, apex trees are included in family F_{rp} (Figures 10a and 10b). Our strategy is called RP-REPOSITION (Algorithm 3).



(a) Apex tree G (it becomes a tree if s is removed) (b) Recursive partition H_G with four (s,t)-paths

Figure 10: Apex tree G and its associated hypergraph H_G.

1: **Input:** hypergraph H_G
2: **if** *there is a single* (s,t)*-path* **then**
3: | the traveller traverses it, returns to s if he is blocked
else
4: | Let $T_1, \ldots, T_q$ be the partition covering hypergraph H_G^+;
5: | Choose one of these sets uniformly: T_i
6: | RP-REPOSITION(T_i)
7: | **if** *all paths of* T_i *are blocked* **then** RP-REPOSITION($\{T_j : j \neq i\}$) **endif**
8: **endif**

Algorithm 3: The RP-REPOSITION strategy

We use the notation from Eq. (9). Our strategy chooses uniformly one hyperedge T_i from the first "sub-list" of H_G (line 5). If T_i contains a single (s,t)-path, the traveller traverses it until t if he can and comes back to s if he is blocked. Otherwise, RP-REPOSITION is applied recursively to the chosen hyperedge T_i: it selects a hyperedge $T_{i,j}$ (from the next level of sub-lists), and so on (line 6). If the traveller does not reach t in the execution of RP-REPOSITION on T_i, we apply RP-REPOSITION to one hyperedge selected randomly among $\{T_j : j \neq i\}$ (line 7). We prove that RP-REPOSITION strategy is $(k+1)$-competitive for (s,t)-paths with almost equal costs.

Theorem 6 *For any graph $G \in F_{rp}$ such that the cost of any simple (s,t)-path P_i is bounded, $\omega_{min} \leq \omega_i \leq (1+\alpha)\,\omega_{min}$, the competitive ratio of* RP-REPOSITION *is $(1+\alpha)\,(k+1)$.*

Proof. For any set T_i in $\{T_1, \ldots, T_q\}$, k_i signifies the number of blockages in it. We suppose, without loss of generality, that sets $T_1, \ldots, T_{q_1}$ are completely blocked with $q_1 < q$ and that there is always an open path in any set $T_{q_1+1}, \ldots, T_q$. With no blocked edge, the competitive ratio of this strategy is 1. We put by induction that the competitive ratio of RP-REPOSITION is $k'+1$ for any graph with $k' < k$ blockages.

We suppose that the first chosen set T_i is completely blocked, *i.e.* $1 \leq i \leq q_1$. In this case, the traveller traverses at most a distance $2k_i\,(1+\alpha)\,\omega_{\min}$, then walks on a subgraph with $k - k_i$ blockages, and traverses a distance shorter than $(k - k_i + 1)\,(1+\alpha)\,\omega_{\min}$. On the contrary, if $i > q_1$, there is an open (s,t)-path in T_i and the distance is at most $(k_i+1)\,(1+\alpha)\,\omega_{\min}$. From $\omega_{\min} \leq \omega_{\text{opt}}$, we determine an upper bound of the competitive ratio of RP-REPOSITION strategy, noted $c_{\mathrm{RP},F_{\mathrm{rp}}}$.

$$c_{\mathrm{RP},F_{\mathrm{rp}}} \leq (1+\alpha)\left(\sum_{i=1}^{q_1}\frac{1}{q}\left(2k_i+(k-k_i)+1\right)+\sum_{i=q_1+1}^{q}\frac{1}{q}\left(k_i+1\right)\right)=(1+\alpha)\left(\frac{q_1}{q}k+\frac{1}{q}\sum_{i=1}^{q}k_i+1\right).$$

As $T_1,\ldots,T_q$ is a partition of H_G, $k=\sum_{i=1}^{q}k_i$. Consequently, $c_{\mathrm{RP},F_{\mathrm{rp}}} \leq (1+\alpha)(k+1)$. ■

References

[1] R. K. Ahuja, Magnanti T. L., and Orlin J. B. *Network flows - theory, algorithms and applications*. Prentice Hall, 1993.

[2] A. Bar-Noy and B. Schieber. The Canadian Traveller Problem. In *Proc. of ACM/SIAM SODA*, pages 261–270, 1991.

[3] M. Bender and S. Westphal. An optimal randomized online algorithm for the k-Canadian Traveller Problem on node-disjoint paths. *J. Comb. Optim.*, 30(1):87–96, 2015.

[4] C. Berge. *Hypergraphes: Combinatoires des ensembles finis*. Gauthier-Villars, 1987.

[5] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge Univ. Press, 1998.

[6] E. D. Demaine, Y. Huang, C. Liao, and K. Sadakane. Canadians Should Travel Randomly. *Proc. of ICALP*, pages 380–391, 2014.

[7] D. R. Ford, L. R.; Fulkerson. Maximal flow through a network. *Canadian Jour. of Math.*, (8):399–404, 1956.

[8] J. M. Kleinberg and E. Tardos. Disjoint Paths in Directed and Undirected graphs. In *Algorithm Design*. Pearson, Addison Wesley edition, 2005.

[9] K. Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicæ*, 10(1):96–115, 1927.

[10] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Math. and Its Applications. OUP Oxford, 2006.

[11] C. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theor. Comput. Sci.*, 84(1):127–150, 1991.

[12] S. Westphal. A note on the k-Canadian Traveller Problem. *Inform. Proces. Lett.*, 106(3):87–89, 2008.

[13] Y. Xu, M. Hu, B. Su, B. Zhu, and Z. Zhu. The Canadian traveller problem and its competitive analysis. *J. Comb. Optim.*, 18(2):195–205, 2009.