Improved Deterministic Strategy for the Canadian Traveller Problem Exploiting Small Max-(s,t)-Cuts

Pierre Bergé[1], Lou Salaün[2,3]

[1] LRI, Université Paris-Sud, Université Paris-Saclay, Orsay France
`Pierre.Berge@lri.fr`
[2] Bell Labs, Nokia Paris-Saclay, 91620 Nozay, France
`lou.salaun@nokia-bell-labs.com`
[3] LTCI, Telecom ParisTech, Université Paris-Saclay, 75013 Paris, France

Abstract. The k-Canadian Traveller Problem consists in finding the optimal way from a source s to a target t on an undirected weighted graph G, knowing that at most k edges are blocked. The traveller, guided by a strategy, sees an edge is blocked when he visits one of its endpoints. A major result established by Westphal is that the competitive ratio of any deterministic strategy for this problem is at least $2k+1$. REPOSITION and COMPARISON strategies achieve this bound.
We refine this analysis by focusing on graphs with a maximum (s,t)-cut size $\mu_{\max}$ less than k. A strategy called DETOUR is proposed and its competitive ratio is $2\mu_{\max} + \sqrt{2}(k - \mu_{\max}) + 1$ when $\mu_{\max} < k$ which is strictly less than $2k+1$. Moreover, when $\mu_{\max} \geq k$, the competitive ratio of DETOUR is $2k+1$ and is optimal. Therefore, DETOUR improves the competitiveness of the deterministic strategies known up to now.

Keywords: Canadian traveller problem · competitive analysis · online algorithms

1 Introduction

Related work. The k-*Canadian Traveller Problem* (k-CTP) was defined by Papadimitriou and Yannakakis [8] and is PSPACE-complete [1, 8]. Given an undirected weighted graph G and two of its vertices $s, t \in V$, the objective is to make a traveller walk from s to t on graph G in the most efficient way despite the existence of some blocked edges $E_* \subsetneq E$. Parameter k is an upper bound of the number of blocked edges: $|E_*| \leq k$. The traveller does not know which edges are blocked when he begins his walk. He discovers a blocked edge $e = (u, v)$ when he visits one of its endpoints u or v.

The traveller traverses graph $G = (V, E, \omega)$, where $n = |V|$ and $m = |E|$. Edge weights are given by the function $\omega : E \to \mathbb{Q}^+$. Our objective is to make the traveller reach target t with a minimum cost (also called distance), which is the sum of the weights of edges traversed. A pair (G, E_*) is called a *road map*.

All the road maps considered are feasible: there is an (s,t)-path in $G\backslash E_*$, the graph G deprived of the obstructed edges E_*.

A solution to the k-CTP is an online algorithm, called a *strategy*. Its quality can be assessed with competitive analysis [4]. Roughly speaking, the competitive ratio is the quotient between the distance actually traversed by the traveller and the distance he would have traversed, knowing which edges are blocked before beginning his walk. Westphal [10] proved that no deterministic strategy achieves a competitive ratio better than $2k+1$. Said differently, for any deterministic strategy A, there is at least one k-CTP road map for which the competitive ratio of A is at least $2k+1$.

Two strategies proposed in the literature reach this optimal ratio: REPOSITION [10] and COMPARISON [11]. REPOSITION makes the traveller traverse the shortest (s,t)-path. If there is a blocked edge (u,v) on this path, the traveller discovers it when he visits vertex u. Then, he comes back to s passing through the same path. The process starts again on $G\backslash E'_*$, the graph G deprived of the blocked edges E'_* identified until now. COMPARISON is based on a different principle: when the traveller discovers a blockage (u,v) and stands on vertex u, he compares the shortest (u,t)-path $P_{\min}^{(u,t)}$ (cost $\omega_{\min}^{(u,t)}$) of $G\backslash E'_*$ with its shortest (s,t)-path $P_{\min}^{(s,t)}$ (cost $\omega_{\min}^{(s,t)}$). If $\omega_{\min}^{(s,t)} \leq \omega_{\min}^{(u,t)}$, the traveller moves as in REPOSITION. If $\omega_{\min}^{(s,t)} > \omega_{\min}^{(u,t)}$, the traveller traverses the path $P_{\min}^{(u,t)}$, etc.

Randomized strategies, *i.e.* strategies in which choices of direction depend on a random draw, were also studied. Westphal [10] proved that there is no randomized strategy achieving a ratio lower than $k+1$. Bender *et al.* [2] studied graphs composed only of vertex-disjoint (s,t)-paths and proposed a polynomial-time strategy of ratio $k+1$. A slight revision of that strategy is reported in [9]. To the best of our knowledge, there is no polynomial-time randomized strategy achieving a competitive ratio smaller than $2k+1$ on general graphs. Such a strategy would not be memoryless [3].

Contributions. Our work exclusively concerns deterministic strategies. We establish a relationship between the size $\mu_{\max}$ of the largest minimal (s,t)-cut of a graph G and the competitive ratio that can be obtained on G, for any configuration of blocked edges. Concretely, the competitive ratio of deterministic strategies on graphs where $\mu_{\max} < k$ is studied.

According to the proof of Lemma 2.1 in [10], for any value $\mu \in \mathbb{N}^*$, there is at least one graph (made up of vertex-disjoint (s,t)-paths only) such that $\mu_{\max} = \mu$ and no deterministic strategy has a competitive ratio less than $2k+1$ on it if $\mu_{\max} \geq k$. In this study, we focus on graphs fulfilling $\mu_{\max} < k$: we assess the competitive ratio of strategies REPOSITION and COMPARISON under this condition. We devise a more competitive strategy called DETOUR. We list our contributions:

- For any value $\mu_{\max} \geq 4$, we prove that there is at least one graph with $\mu_{\max} < k$ for which both REPOSITION/COMPARISON strategies are $(2k+1)$-competitive.
- We propose a polynomial-time strategy called DETOUR with competitive ratio $2\mu_{\max}+\sqrt{2}(k-\mu_{\max})+1$ when $\mu_{\max} < k$. It outperforms the competitive

ratio of the existing deterministic strategies. In brief, ratio $2k+1$ is widely defeated by a deterministic strategy on graphs G satisfying $\mu_{\max} < k$.

Strategy DETOUR is also $(2k+1)$-competitive when $\mu_{\max} \geq k$. For this reason, it becomes the best deterministic strategy known for the k-CTP because it performs as well as REPOSITION/COMPARISON when $\mu_{\max} \geq k$ and better than them when $\mu_{\max} < k$.

The organization of this article follows. In Section 2 we remind some definitions related to online algorithms, paths and cuts. Section 3 contains the proof that REPOSITION and COMPARISON are $(2k+1)$-competitive, even if $\mu_{\max} < k$. The DETOUR strategy is described in Section 4 and its competitive ratio is evaluated. We conclude this study in Section 5 and provide some directions for future research.

2 Preliminaries

We present the definition of the competitive ratio and some notions associated with paths and cuts.

Competitive ratio. For any set of blocked edges $E'_* \subseteq E_*$, $\omega_{\min}(G, E'_*)$ is the cost of the shortest (s,t)-path in graph $G \backslash E'_*$. Value $\omega_{\text{opt}} = \omega_{\min}(G, E_*)$ is the *optimal offline cost* for the road map (G, E_*). Concretely, this corresponds to the distance the traveller would have traversed if he had known the blockages in advance.

The competitive ratio is defined in [4]. We denote by $\omega_A(G, E_*)$ the distance traversed by the traveller guided by strategy A on graph G from source s to target t with blocked edges E_*. The competitive ratio $c_A(G, E_*)$ of A over a road map (G, E_*) is defined as $c_A(G, E_*) = \frac{\omega_A(G,E_*)}{\omega_{\text{opt}}}$. The competitive ratio c_A of A is thus:

$$c_A = \max_{(G,E_*)} c_A(G, E_*) \tag{1}$$

Similarly, we say strategy A is c_A-competitive for a family $\mathcal{F}$ of graphs (for example, $\mathcal{F} = \{G : \mu_{\max} < k\}$) if it is the maximum of value $c_A(G, E_*)$ over road maps (G, E_*) such that $G \in \mathcal{F}$.

Paths. A *simple path* P is a sequence of pairwise different vertices $v_1 \cdot v_2 \cdots v_i \cdot v_{i+1} \cdots v_\ell$, with departure v_1 and arrival v_ℓ, such that two successive vertices (v_i, v_{i+1}) are adjacent in G. All paths mentioned in this article are simple. To improve readability, we abuse notations: $v_1 \in P$ and $(v_1, v_2) \in P$ mean that vertex v_1 and edge (v_1, v_2) are on path P, respectively. If vertices u and v belong to path P, then $P^{(u,v)}$ denotes the section of path P between vertices u and v. Any path is naturally associated with a direction, from the departure to the arrival. We say the *successor* of edge e in P is the edge arriving just after e in P. The *descendants* of e are all edges arriving after e in P, *i.e.* edges further than u from the departure of P. The *predecessor* and the *ancestors* are defined symmetrically.

Graphs may contain several shortest (s,t)-paths. Our algorithm in Section 4 requires to compute one of the shortest (s,t)-paths of any graph in a deterministic way. To achieve it, a solution is to associate any vertex with an identifier in $\{1,\dots,n\}$. If two paths have the same distance, we compare their lexicographic order. Dijkstra's algorithm [6] is adapted to this extra criterion: for any vertex v, it stores the shortest path from the start point to v with the smallest lexicographic order. Whenever we refer to "the shortest (u,v)-path", for any vertices u and v, this process is executed.

Cuts. A set $X \subseteq E$ is an edge (s,t)-cut if source s and target t are separated in graph G deprived of edges X. We say that cut X is *minimal* if none of its proper subsets $X' \subsetneq X$ is an (s,t)-cut. Let $\mu_{\max}$ be the maximum cardinality of a minimal (s,t)-cut:

$$\mu_{\max} = \max_{\substack{X \text{ minimal} \\ (s,t)-\text{cut}}} |X| \,. \tag{2}$$

Any (s,t)-cut X where $|X| > \mu_{\max}$ is not minimal. If X is a minimal (s,t)-cut, graph $G\backslash X$ contains exactly two connected components: one, denoted $R(X,s)$, contains all vertices reachable from s and another one, denoted $R(X,t)$, all vertices reachable from t. Largest minimal (s,t)-cuts $X_{\max}$, $|X_{\max}| = \mu_{\max}$, are called max-(s,t)-cuts throughout our study.

3 Competitive ratio of existing strategies when $\mu_{\max} < k$

We study the family of graphs satisfying $\mu_{\max} < k$. We assess, on such instances, the competitiveness of the two best deterministic strategies known for now in the literature. Indeed, REPOSITION and COMPARISON are $(2k+1)$-competitive for general graphs. We prove that they do not benefit from the inequality $\mu_{\max} < k$. We begin with REPOSITION strategy.

Theorem 1. *For any $k > 4$, there is a road map $(G_k, E_{*,k})$, $\mu_{max} = 4$, such that the competitive ratio of* REPOSITION *on $(G_k, E_{*,k})$ is $2k+1$: $c_{\text{rep}}(G_k, E_{*,k}) = 2k+1$.*

Proof. The road map $(G_k, E_{*,k})$ is drawn in Fig. 1. Graph G_k has a horizontal axis of symmetry Δ. On each side, there are $\lceil \frac{k}{2} \rceil$ diamond graphs, *i.e.* cycles of length 4, put in series. They are surrounded by two edges, one of weight 1 incident to s and one of weight $\varepsilon \ll 1$ incident to t. For any diamond graph above Δ, three of its edges are weighted with ε and the bottom left one is weighted with 3ε. All the top right edges are blocked (red edges in Fig. 1). All diamonds below Δ are identical, except for the one closest to s (weights 2ε, ε, 4ε, and ε, see Fig. 1). If k is even, as in Fig. 1, the top right edges of all diamonds are blocked. If k is odd, there is no blockage on the diamond below Δ which is the closest to t. In this way, there are always k blocked edges in $E_{*,k}$ and the max-(s,t)-cut size of G_k is $\mu_{\max} = 4$. Let $g(k) = 2\lceil \frac{k}{2} \rceil \in \{k, k+1\}$. The cost of the shortest (s,t)-path in G_k is $1 + (g(k)+1)\varepsilon$.

Guided by REPOSITION, the traveller traverses the shortest (s,t)-path which is above Δ and is blocked in the first diamond (distance $1+\varepsilon$). Set E'_* denotes the

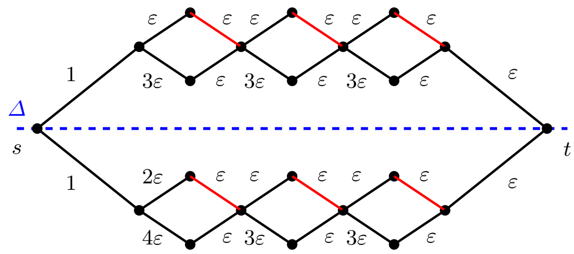

Fig. 1: Graph G_6 and blocked edges $E_{*,6}$ in red

blocked edges discovered during the execution: for now, $|E'_*| = 1$. The traveller comes back to s (distance $1+\varepsilon$). The shortest (s,t)-path in graph $G\backslash E'_*$ is now below axis Δ and its cost is $1 + (g(k)+2)\varepsilon$ as it contains an edge of weight 2ε. The traveller traverses this path and is blocked in the first diamond below Δ (distance $1+2\varepsilon$). Then, the current shortest (s,t)-path in $G\backslash E_*$ is above Δ and its cost is $1+(g(k)+3)\varepsilon$, etc. In summary, the traveller is blocked k times traversing paths with cost larger than $1+\varepsilon$ in two directions. The total distance traversed d_{rep} satisfies $d_{\text{rep}} \geq 2k\,(1+\varepsilon) + \omega_{\text{opt}} \geq (2k+1)(1+\varepsilon)$. As $\omega_{\text{opt}} = 1 + (2g(k)+1)\varepsilon$, the competitive ratio of REPOSITION c_{rep} is thus:

$$c_{\text{rep}} \geq (2k+1)\frac{1+\varepsilon}{1+(2g(k)+1)\varepsilon} \overset{\varepsilon\to 0}{\longrightarrow} 2k+1.$$

As ε may tend to zero, there always is a road map on which REPOSITION achieves a ratio $2k+1-\delta$ for any arbitrarily small value $\delta > 0$.

This result remains true for any value $\mu_{\max} > 4$ as we can artificially add (s,t)-paths disjoint from G_k which make $\mu_{\max}$ increase. It suffices to assign a sufficiently large cost to these paths, so that REPOSITION never makes the traveller traverse them. Now we focus on COMPARISON strategy.

Theorem 2. *For any $k > 3$, there is a road map $(G'_k, E'_{*,k})$, $\mu_{max} = 3$, such that the competitive ratio of* COMPARISON *on $(G'_k, E'_{*,k})$ is $2k+1$: $c_{\text{comp}}(G'_k, E'_{*,k}) = 2k+1$.*

Proof. Road map $(G'_k, E'_{*,k})$ is drawn in Fig. 2. Axis Δ' is represented to facilitate the description of G'_k. Above Δ', $k-1$ diamonds graphs are put in series and are surrounded as in G_k (see Theorem 1). On each diamond, the edge weights are ε, except for the bottom left edges weighted with value 1. The top left edges are blocked. Moreover, the edge incident to t above Δ' is also blocked, so $\left|E'_{*,k}\right| = k$. Below Δ', there is an open (s,t)-path with cost $1+2k\varepsilon$. The shortest (s,t)-path in G'_k is above Δ' and its cost is $1+(2k-1)\varepsilon$. Graph G'_k is such that $\mu_{\max} = 3$.

Guided by COMPARISON, the traveller traverses the shortest (s,t)-path and is blocked when he arrives on the first diamond (distance 1). Then, the cost of the shortest (s,t)-path in $G\backslash E'_*$, i.e. $1+2k\varepsilon$, is compared with the shortest

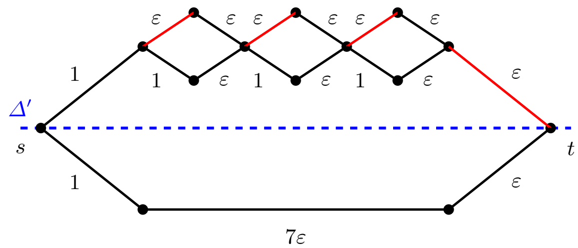

Fig. 2: Graph G'_4 and blocked edges $E'_{*,4}$ in red

distance between the current position of the traveller and t, i.e. $1 + (2k - 2)\varepsilon$. Since $1 + (2k - 2)\varepsilon < 1 + 2k\varepsilon$, the traveller chooses to take the shortest path between its current position and t, which is above Δ'. He meets a second blockage when arriving on the second diamond (distance $1 + \varepsilon$). Then, he makes the same decision and traverses the diamonds above Δ'. Eventually, when he meets the last blockage incident to t, he travels back to s and finally passes through the optimal offline path, below Δ'. The total distance traversed is $d_{\text{comp}} = 2 + 2(k - 1)(1 + \varepsilon) + 1 + 2k\varepsilon$. The competitive ratio c_{comp} of COMPARISON strategy on the road map $(G'_k, E'_{*,k})$ follows:

$$c_{\text{comp}} = \frac{2 + 2(k-1)(1+\varepsilon) + 1 + 2k\varepsilon}{1 + 2k\varepsilon} \xrightarrow{\varepsilon \to 0} 2k + 1.$$

Making ε tend to zero terminates the proof.

The existence of a deterministic strategy achieving a ratio less than $2k+1$ on graphs fulfilling $\mu_{\max} < k$ is still an open question after the results established in Theorems 1 and 2. Indeed, we showed that the existing strategies cannot defeat their global competitive ratio on this particular family of graphs. In the remainder, we devise a strategy outperforming REPOSITION and COMPARISON when $\mu_{\max} < k$.

4 Detour strategy

We first introduce in Subsection 4.1 a parameterized strategy called α-DETOUR. It takes as input graph G, source s, target t, and a parameter $\alpha \in (0, 1)$. In Subsection 4.2, we provide an upper bound of its competitive ratio. This bound is minimized for $\alpha = \frac{\sqrt{2}}{2}$ and is $2\mu_{\max} + \sqrt{2}(k - \mu_{\max}) + 1$ in this case. Strategy DETOUR mentioned earlier corresponds to $\frac{\sqrt{2}}{2}$-DETOUR. Finally, we provide the execution time of DETOUR strategy and discuss some of its properties in Subsection 4.3.

4.1 Description of α-Detour strategy

We present the α-DETOUR strategy in Algorithm 1. Variable pos keeps track of the traveller's current position. The idea is to perform successively two phases:

an *exploration* followed by a *detour-backtracking.* The exploration starts when the traveller is on source s (line 8). He traverses the shortest (s,t)-path $P_{\min}^{(s,t)}$ called the *exploration path.* Its cost $\omega_{\min}^{(s,t)} = \omega_{\min}(G, E'_*)$ is stored in $\omega_{\exp}$ (line 7). At this point, there are two possibilities:
1) The traveller reaches t and the execution terminates (line 13).
2) The traveller arrives at $\mathsf{pos} = u$ and discovers a blocked edge $(u,v) \in P_{\min}^{(s,t)}$. Then, the detour-backtracking phase begins.

Each exploration followed by a detour-backtracking phase can be seen as a depth-first search (DFS). When the traveller is blocked on $P_{\min}^{(s,t)}$, we ask whether an α-*detour*, *i.e.* a (pos, t)-path with cost at most $\alpha\omega_{\exp}$, exists. If an α-detour exists, the traveller traverses the shortest path $P_{\min}^{(\mathsf{pos},t)}$ from the current position pos to target t (line 9). Obviously, its cost satisfies $\omega_{\min}^{(\mathsf{pos},t)} \leq \alpha\omega_{\exp}$. Otherwise, the traveller backtracks to the vertex before $\mathsf{pos} = u$ on the exploration path (lines 14-16).

As in a DFS, we use a stack to remember the previous vertices for backtracking. We denote by V_{stack} the set of vertices in the stack. We do not allow an α-detour $P_{\min}^{(\mathsf{pos},t)}$ to pass through any vertex $v \in V_{\text{stack}}$, since the section $P_{\min}^{(v,t)}$ will be considered later on when $\mathsf{pos} = v$. The vertices of an exploration path traversed by the traveller are naturally put in stack. Moreover, when the traveller is blocked on an α-detour $P_{\min}^{(\mathsf{pos},t)}$, the vertices of $P_{\min}^{(\mathsf{pos},t)}$ from pos to the endpoint of the blocked edge are put in stack. Finally, if the traveller backtracks to s, the algorithm goes back to the exploration phase. At this moment, the stack is empty.

Recall that E'_* represents the set of discovered blocked edges. Variable G' contains the graph G deprived of the discovered blockages E'_* at any moment of the execution. At each iteration of the while loop, the variables are updated as follows: if the path $P_{\min}^{(u_0,t)}$ currently traversed (lines 8-9) does not contain any blockage, then the traveller reaches t, i.e. $\mathsf{pos} \leftarrow t$. In this case, the algorithm terminates since the destination is reached. Otherwise, let $P_{\min}^{(u_0,t)} = u_0 \cdots u_i \cdot u_{i+1} \cdots u_r \cdot t$, where (u_i, u_{i+1}) is its first blocked edge. The traveller's position is updated from u_0 to u_i (line 10). Then, we update E'_* with the newly discovered blockages including (u_i, u_{i+1}), and $G' \leftarrow G \backslash E'_*$ (line 12). In addition, we push the traversed vertices $u_0, \ldots, u_{i-1}$ on the stack (except u_i) and update accordingly $V_{\text{stack}} \leftarrow V_{\text{stack}} \cup \{u_0, \ldots, u_{i-1}\}$. In case there is no α-detour $P_{\min}^{(u_i,t)}$ in $G' \backslash V_{\text{stack}}$, the algorithm backtracks by popping u_{i-1} from the stack and setting $\mathsf{pos} \leftarrow u_{i-1}$ (lines 14-16).

If $\alpha = 0$, the algorithm does not take any detour. As a consequence, 0-DETOUR is equivalent to REPOSITION, as both procedures perform an exploration phase followed by backtracking without taking any detour. In the following, we provide an upper bound of α-DETOUR's competitive ratio.

Algorithm 1: The α-DETOUR strategy

Input: graph G, source s, target t, parameter $\alpha \in (0,1)$
$E'_* \leftarrow \emptyset$; $G' \leftarrow G \backslash E'_*$; $\mathsf{pos} \leftarrow s$; $u_0 \leftarrow s$; $\omega_{\text{exp}} \leftarrow \omega_{\text{min}}(G, \emptyset)$;
$\mathsf{stack} \leftarrow \mathsf{Empty\ Stack}$; $V_{\text{stack}} \leftarrow \emptyset$;
while true **do**
  $u_0 \leftarrow \mathsf{pos}$;
  if $u_0 = s$ **then**
    $\omega_{\text{exp}} \leftarrow \omega_{\text{min}}(G, E'_*)$;
    traverse the shortest (s,t)-path $P_{\text{min}}^{(u_0,t)}$ in G';
  else
    traverse the shortest (u_0,t)-path $P_{\text{min}}^{(u_0,t)}$ in $G' \backslash V_{\text{stack}}$;
  endif
  update pos;
  push the vertices visited in $P_{\text{min}}^{(u_0,t)}$ except pos on stack;
  update E'_*, G', and V_{stack};
  if $\mathit{pos} = t$ **then break**;
  while $\mathit{pos} \neq s$ **and** *there is no* $P_{min}^{(pos,t)}$ *in* $G' \backslash V_{stack}$ *such that* $\omega_{min}^{(pos,t)} \leq \alpha \omega_{exp}$ **do**
    $\mathsf{pos} \leftarrow \text{pop}(\mathsf{stack})$;
    $V_{\text{stack}} \leftarrow V_{\text{stack}} \backslash \{\mathsf{pos}\}$;
  end
end

4.2 Competitive analysis

We denote by $P_1, \ldots, P_\ell$ the exploration paths $P_{\text{min}}^{(s,t)}$ such that the distance from s to the blocked edge discovered on it is greater than α multiplied by their own cost, *i.e.* $\alpha\omega_i$. In other words, the distance d_i traversed by the traveller on the exploration paths P_i, $1 \leq i \leq \ell$, satisfies $d_i \geq \alpha\omega_i$. Paths P_i are sorted in order to fulfil $\omega_1 \leq \cdots \leq \omega_\ell$. The exploration paths $P_1, \ldots, P_{\ell-1}$ are blocked, while path P_ℓ can be open. If P_ℓ does not contain any blockage, then the algorithm terminates after the traveller traverses it.

Let us partition $P_1, \ldots, P_\ell$ into two sequences $S_1 = P_1, \ldots, P_{h-1}$ and $S_2 = P_h, \ldots, P_\ell$ such that $2\alpha\omega_{h-1} < \omega_\ell \leq 2\alpha\omega_h$. In the particular case where $\omega_\ell \leq 2\alpha\omega_1$, then $h = 1$ and the two sets are $S_1 = \emptyset$ and $S_2 = P_1, \ldots, P_\ell$. We denote by $G[P_h, \ldots, P_\ell]$ the subgraph of G induced by paths $P_h, \ldots, P_\ell$, *i.e.* containing only the vertices and edges of paths P_i, $h \leq i \leq \ell$.

Theorem 3. *The max-(s,t)-cut size induced on graph $G[P_h, \ldots, P_\ell]$ is at least $\ell - h + 1$.*

Proof. We denote by b_i the blocked edge discovered on P_i, for $i \in \{h, \ldots, \ell\}$. We construct inductively a set $\{e_h, \ldots, e_\ell\}$ of edges satisfying the following induction hypotheses, for all $i \in \{h, \ldots, \ell\}$:

$H_1(i)$: $\{e_h, \ldots, e_i\}$ is a minimal (s,t)-cut of $G[P_h, \ldots, P_i]$,

$H_2(i)$: Either $e_i = b_i$ or e_i is an ancestor of b_i in P_i,
$H_3(i)$: For $j \in \{i+1, \ldots, \ell\}$, P_j cannot pass through e_i.

Basis: For $i = h$, $G[P_h, \ldots, P_i]$ contains only one path P_h. We choose $e_h = b_h$, which fulfils $H_2(h)$. Since any edge of P_h is a max-(s,t)-cut of $G[P_h]$, it satisfies $H_1(h)$. Statement $H_3(h)$ is also true, as e_h is blocked.

Inductive step: Assume that $H_1(i)$ to $H_3(i)$ are true for a certain integer i in $\{h, \ldots, \ell-1\}$. We will construct e_{i+1} and prove the induction hypotheses $H_1(i+1)$ to $H_3(i+1)$. For simplicity, we denote sets $R(\{e_h, \ldots, e_i\}, s)$ and $R(\{e_h, \ldots, e_i\}, t)$ in graph $G[P_h, \ldots, P_i]$ by $R_i(s)$ and $R_i(t)$, respectively.

Let $P_{i+1}^{(v_0,v_p)} = v_0 \cdot v_1 \cdots v_p$ be the longest section in P_{i+1}, starting from $v_0 = s$, such that $v_0, \ldots, v_p \in R_i(s)$ and $p \in \mathbb{N}$. Section $P_{i+1}^{(v_0,v_p)}$ contains at least vertex $v_0 = s$. For $j \in \{h, \ldots, i\}$, all ancestors of e_j in P_j belong to $R_i(s)$, and all descendants belong to $R_i(t)$. Therefore, according to $H_2(i)$, all exploration paths' sections of the form $P_j^{(s,u)}$ are open and equal to the shortest path from s to u, for $u \in R_i(s) \cap P_j$ and $j \in \{h, \ldots, i\}$. In particular, since $P_{i+1}^{(v_0,v_p)}$ is the shortest (v_0, v_p)-path, we deduce that it is open as v_p belongs to some P_j by definition of $R_i(s)$.

According to $H_3(i)$, $P_{i+1}^{(v_p,t)}$ is a new path connecting $R_i(s)$ to $R_i(t)$, which does not traverse any edge of the cut $\{e_h, \ldots, e_i\}$. Furthermore, no vertex in $P_{i+1}^{(v_{p+1},t)}$ belongs to $R_i(s)$. Indeed, suppose for the sake of contradiction that $u \in P_{i+1}^{(v_{p+1},t)}$ and $u \in R_i(s)$. There would exist $j \in \{h, \ldots, i\}$, such that $P_j^{(s,u)}$ is the shortest (s,u)-path, and all its vertices belong to $R_i(s)$. This contradicts with the fact that $P_{i+1}^{(s,u)}$ is also the shortest (s,u)-path and $v_{p+1} \notin R_i(s)$, by definition. Let $v_{p'}$ be the first vertex of P_{i+1} belonging to $R_i(t)$, *i.e.* $v_{p'} \in R_i(t)$ and $p < p'$. Such a vertex exists as t is a candidate. We derive that $P_{i+1}^{(v_p,v_{p'})}$ is the unique path both connecting $R_i(s)$ to $R_i(t)$ and avoiding cut X. Figure 3 represents cut $\{e_h, \ldots, e_i\}$, path P_{i+1} and its vertices v_p and $v_{p'}$.

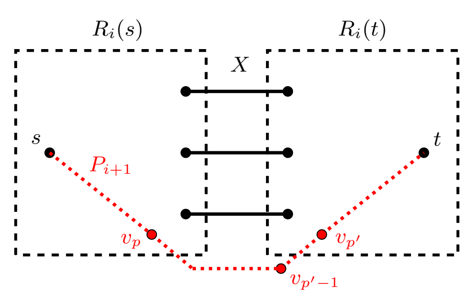

Fig. 3: Cut $X = \{e_h, \ldots, e_i\}$, path P_{i+1}, and vertices v_p, $v_{p'-1}$, $v_{p'}$

We fix e_{i+1} differently depending on the position of b_{i+1}. We already proved that $b_{i+1} \notin P_{i+1}^{(v_0,v_p)}$, the remaining cases are:

- If $b_{i+1} \in P_{i+1}^{(v_p, v_{p'})}$, then we set $e_{i+1} = b_{i+1}$. As $e_{i+1} \in E_*$, $H_3(i+1)$ is true.
- Otherwise, if $b_{i+1} \in P_{i+1}^{(v_{p'}, t)}$, we choose $e_{i+1} = (v_{p'-1}, v_{p'})$. We prove that the cost of the current shortest $(s, v_{p'})$-path, $P_{i+1}^{(s, v_{p'})}$, is at least $\alpha\omega_h$. Indeed, as vertex $v_{p'}$ belongs to a certain path $P_{j'}$, $j' \in \{h, \dots, i\}$, the cost of $P_{i+1}^{(s, v_{p'})}$ is at least the cost of $P_{j'}^{(s, v_{p'})}$. If we have $\omega_{i+1}^{(s, v_{p'})} \le \alpha\omega_h$, the distance traversed by the traveller on $P_{j'}$ is less than $\alpha\omega_h \le \alpha\omega_{j'}$, as $v_{p'} \in R_i(t)$. This contradicts with the fact that $P_{j'} \in \{P_h, \dots, P_\ell\}$. Moreover, after the $(i+1)$-th detour-backtracking phase, all remaining open $(v_{p'}, t)$-paths are longer than $\alpha\omega_{i+1} \ge \alpha\omega_h$. Therefore, the cost of any exploration (s,t)-path passing through $v_{p'}$ is greater than $\alpha\omega_h + \alpha\omega_{i+1} \ge 2\alpha\omega_h$. This is impossible since the last exploration path P_ℓ satisfies $\omega_\ell \le 2\alpha\omega_h$. As a consequence, no exploration path passes through $v_{p'}$ and $H_3(i+1)$ is true.

Both cases fulfil naturally $H_2(i+1)$. It only remains to prove statement $H_1(i+1)$. We showed that $P_{i+1}^{(v_p, v_{p'})}$ is the only path connecting $R_i(s)$ to $R_i(t)$, and $e_{i+1} \in P_{i+1}^{(v_p, v_{p'})}$. Thus, $\{e_h, \dots, e_{i+1}\}$ is an (s,t)-cut of $G[P_h, \dots, P_{i+1}]$. If we re-open edge e_{i+1}, path $P_{i+1}^{(v_p, v_{p'})}$ connects $R_i(s)$ to $R_i(t)$. If we re-open e_j, $j < i+1$, there is a path in $G[P_h, \dots, P_i]$ which connects $R_i(s)$ to $R_i(t)$ independently of $P_{i+1}^{(v_p, v_{p'})}$, according to the minimality of $\{e_h, \dots, e_i\}$ in $H_1(i)$. As a consequence, no proper subset of $\{e_h, \dots, e_{i+1}\}$ is an (s,t)-cut. Cut $\{e_h, \dots, e_{i+1}\}$ is minimal.

In summary, we derive by induction that $\{e_h, \dots, e_\ell\}$ is a minimal (s,t)-cut of $G[P_h, \dots, P_\ell]$. The size of the max-(s,t)-cut is at least $\ell - h + 1$.

The following lemma states that the max-(s,t)-cut size of graph $G[P_h, \dots, P_\ell]$ cannot exceed the max-(s,t)-cut size of the bigger graph G.

Lemma 1. *The max-(s,t)-cut size on graph $G[P_h, \dots, P_\ell]$ is less than or equal to the max-(s,t)-cut size μ_{max} of the original graph G.*

Proof. Let X be one of the max-(s,t)-cuts in graph $G[P_h, \dots, P_\ell]$. Cut X is minimal, so no subset $X' \subsetneq X$ is an (s,t)-cut. If X is an (s,t)-cut in G, then it is also minimal in G as none of its subsets can be an (s,t)-cut. Therefore, $|X| \le \mu_{\max}$.

Suppose now that X is not an (s,t)-cut in G. We denote by Y the max-(s,t)-cut in graph G deprived of edges X, *i.e.* $G \backslash X$. Set $X \cup Y$ is thus a minimal (s,t)-cut in graph G as Y is minimal in $G \backslash X$. So, $|X| \le |X \cup Y| \le \mu_{\max}$. In both cases, the max-(s,t)-cut size in $G[P_h, \dots, P_\ell]$ is at most $\mu_{\max}$.

According to both Theorem 3 and Lemma 1, a relationship exists between values ℓ, h, and $\mu_{\max}$, which is $\ell - h + 1 \le \mu_{\max}$.

After traversing an exploration path P_i, the traveller performs a detour-backtracking phase. The number of blockages discovered during this i-th detour-backtracking phase is denoted by q_i. We analyse the cost of traversing P_i and performing the i-th detour-backtracking phase in Lemma 2.

Lemma 2. *The total cost of both the i-th exploration phase and the i-th detour-backtracking phase is not greater than $(2 + 2\alpha q_i)\omega_i$.*

Proof. The stack in Algorithm 1 ensures that each edge is only traversed twice: first time when moving towards t on an exploration path or a detour, and a second time when backtracking. The exploration path costs ω_i and each detour costs no more than $\alpha\omega_i$. Besides, the number of detours is at most q_i. Hence, the total cost is at most $2\omega_i + q_i 2\alpha\omega_i$, which concludes the proof.

We denote by k_1 (resp. k_2) the number of blocked edges discovered during the exploration and detour-backtracking phases associated with paths $P_1, \ldots, P_{h-1}$ (resp. $P_h, \ldots, P_\ell$). Let k_3 be the number of blockages discovered during the other phases, so that $k_1 + k_2 + k_3 = k$. We derive in Theorem 4 an upper-bound on the competitive ratio as a function of k_1, k_2, k_3, and α.

Theorem 4. *The competitive ratio of α-DETOUR is upper-bounded by:*

$$\frac{k_1}{\alpha} + 2\mu_{max} + 2\alpha(k_2 + k_3 - \mu_{max}) + 1. \tag{3}$$

Proof. Since path P_ℓ is the shortest (s,t)-path of a certain graph $G \backslash E'_*$ where $E'_* \subseteq E_*$, the offline optimal cost satisfies

$$\omega_{\text{opt}} \geq \omega_\ell. \tag{4}$$

According to Lemma 2, the distance traversed during the exploration and detour-backtracking phases of $P_1, \ldots, P_{h-1}$ is not greater than

$$\sum_{j=1}^{h-1} (2 + 2\alpha q_j)\omega_j \leq 2\omega_{h-1} \sum_{j=1}^{h-1} (1 + q_j) = 2k_1\omega_{h-1}. \tag{5}$$

Inequality (5) comes from the fact that $\omega_1 \leq \cdots \leq \omega_{h-1}$ and $\sum_{j=1}^{h-1} (1 + q_j) = k_1$.

We evaluate the cost of the phases associated with $P_h, \ldots, P_\ell$. Path P_ℓ is either open and traversed in one direction only (Case 1) or it is blocked and the traveller reaches t via a detour (Case 2).

Case 1: If P_ℓ does not contain any blockage, then the algorithm terminates after traversing it. This final exploration phase costs ω_ℓ. We have $q_\ell = 0$ and $k_2 = \sum_{j=h}^{\ell-1} (1 + q_j)$. Given Lemma 2, the cost of the h-th to ℓ-th phases is less than:

$$\begin{aligned}
\sum_{j=h}^{\ell-1} (2 + 2\alpha q_j)\omega_j + \omega_\ell &= \sum_{j=h}^{\ell-1} (2\alpha + 2\alpha q_j)\omega_j + \sum_{j=h}^{\ell-1} (2 - 2\alpha)\omega_j + \omega_\ell, \\
&\leq 2\alpha k_2\omega_\ell + (2 - 2\alpha)(\ell - h)\omega_\ell + \omega_\ell, \qquad (6) \\
&< 2\alpha k_2\omega_\ell + (2 - 2\alpha)\mu_{\max}\omega_\ell + \omega_\ell, \qquad (7) \\
&= 2\alpha(k_2 - \mu_{\max})\omega_\ell + 2\mu_{\max}\omega_\ell + \omega_\ell.
\end{aligned}$$

We deduce Inequality (6) from $\omega_h \leq \cdots \leq \omega_\ell$. By applying Theorem 3 and Lemma 1 on $S_2 = P_h, \ldots, P_\ell$, we derive that $\ell - h \leq \mu_{\max} - 1 < \mu_{\max}$ in Inequality (7).

Case 2: Suppose that P_ℓ is blocked. The ℓ-th exploration and detour-backtracking phases cost at most $(2 + 2\alpha q_\ell)\omega_\ell + \alpha\omega_\ell$. Moreover, we have $k_2 = \sum_{j=h}^{\ell} (1 + q_j)$. The distance traversed from the h-th to the ℓ-th phases is not greater than:

$$\begin{aligned}
\sum_{j=h}^{\ell-1}(2 + 2\alpha q_j)\omega_j + (2 + 2\alpha q_\ell + \alpha)\omega_\ell &= \sum_{j=h}^{\ell} (2 + 2\alpha q_j)\omega_j + \alpha\omega_\ell, \\
&\leq 2\alpha k_2\omega_\ell + (2 - 2\alpha)(\ell - h + 1)\omega_\ell + \alpha\omega_\ell, && (8)\\
&\leq 2\alpha k_2\omega_\ell + (2 - 2\alpha)\mu_{\max}\omega_\ell + \alpha\omega_\ell, && (9)\\
&\leq 2\alpha(k_2 - \mu_{\max})\omega_\ell + 2\mu_{\max}\omega_\ell + \omega_\ell. && (10)
\end{aligned}$$

Inequality (8) follows from $\omega_h \leq \cdots \leq \omega_\ell$. We obtain (9) from $\ell - h + 1 \leq \mu_{\max}$. Finally, $\alpha \leq 1$ implies Eq. (10).

Contrary to $P_1, \ldots, P_\ell$, some exploration paths $\widehat{P}$ may be such that the distance traversed on them is at most α multiplied by their own cost $\widehat{\omega}$. The distance traversed during the phases which are not asssociated with $P_1, \ldots, P_\ell$ is the cost of these exploration paths $\widehat{P}$ and their α-detours. As $\widehat{\omega} \leq \omega_{\text{opt}}$, it is at most $2\alpha k_3\omega_{\text{opt}}$. Applying Eq. (4), the competitive ratio of α-DETOUR admits the following upper-bound:

$$\begin{aligned}
\frac{\omega_{\alpha-\text{DETOUR}}}{\omega_{\text{opt}}} &\leq \frac{2k_1\omega_{h-1} + 2\alpha(k_2 + k_3 - \mu_{\max})\omega_{\text{opt}} + 2\mu_{\max}\omega_{\text{opt}} + \omega_{\text{opt}}}{\omega_{\text{opt}}}, \\
&\leq \frac{k_1\omega_\ell}{\alpha\omega_{\text{opt}}} + 2\mu_{\max} + 2\alpha(k_2 + k_3 - \mu_{\max}) + 1, && (11)\\
&\leq \frac{k_1}{\alpha} + 2\mu_{\max} + 2\alpha(k_2 + k_3 - \mu_{\max}) + 1. && (12)
\end{aligned}$$

Inequality (11) follows from the partition $\{S_1, S_2\}$ which imposes $2\alpha\omega_{h-1} < \omega_\ell$.

Let $c_{\text{det}}(k_1, k_2, k_3, \alpha)$ denote the value in (12). Parameters k_1, k_2, and k_3 depend on the road map (G, E_*), so only $\alpha \in (0, 1)$ can be tuned. Value $\alpha = \frac{\sqrt{2}}{2}$ minimizes $c_{\text{det}}(k_1, k_2, k_3, \alpha)$ under the condition $k_1 + k_2 + k_3 = k$ for any $k > \mu_{\max}$. Formally,

$$\frac{\sqrt{2}}{2} = \operatorname*{argmin}_{0 \leq \alpha \leq 1} \max_{\substack{k_1, k_2, k_3 \in \mathbb{N} \\ k_1 + k_2 + k_3 = k}} c_{\text{det}}(k_1, k_2, k_3, \alpha)$$

Corollary 1. *The competitive ratio of* DETOUR *is at most* $2\mu_{max} + \sqrt{2}(k - \mu_{max}) + 1$.

Proof. We obtain this ratio by setting $\alpha = \frac{\sqrt{2}}{2}$ and $k_1 + k_2 + k_3 = k$ in Eq. (3).

4.3 Discussion

In summary, strategy DETOUR is as competitive as REPOSITION/COMPARISON for the range $\mu_{\max} \geq k$ but more competitive for the range $1 \leq \mu_{\max} < k$. The slope of the competitive ratio of DETOUR when k varies is only $\sqrt{2}$ for $\mu_{\max} < k$. Figure 4 gives the shape of the competitive ratios of REPOSITION (in blue) and DETOUR (in red) as a function of k.

DETOUR strategy needs to identify the shortest (s,t)-paths and (pos,t)-paths at any moment of its execution. To achieve it, Dijkstra's algorithm [6] is computed once between two discoveries of blocked edges with t as the start point. Hence, the running time of DETOUR is $O(k(m + n \log n))$.

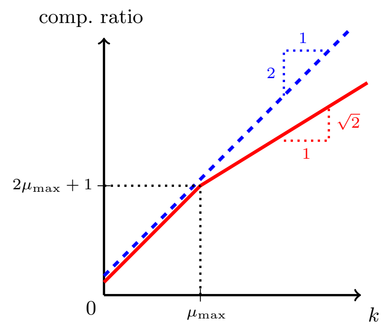

Fig. 4: Competitiveness of REPOSITION (blue) and DETOUR (red) versus k

As for REPOSITION and COMPARISON, the execution of DETOUR strategy is independent of the value of k. Thus, it can be used when no upper bound on the number of blockages is known and its competitive ratio is $2\mu_{\max} + \sqrt{2}(|E_*| - \mu_{\max}) + 1$.

DETOUR strategy can be executed without knowing the value $\mu_{\max}$. Indeed, the competitive ratio of DETOUR depends on $\mu_{\max}$ but no decision is made based on $\mu_{\max}$ in Algorithm 1. In the next paragraph, we explain that value $\mu_{\max}$ cannot be computed in polynomial time.

Finding one of the largest minimal (s,t)-cuts $X_{\max}$, $|X_{\max}| = \mu_{\max}$ is a NP-hard problem, even for planar graphs [7]. A linear time algorithm computing $\mu_{\max}$ exists only for series-parallel graphs [5]. In summary, it is not possible to evaluate value $\mu_{\max}$ for any graph in polynomial time, assuming P≠NP. Given an input graph G, the competitive ratio of DETOUR strategy on any road map (G, E_*) cannot be predicted fast. The only possibility is thus to execute DETOUR, which runs in polynomial time, on G directly and then to verify whether a gain of competitiveness is obtained.

5 Conclusion

Even if the global competitiveness of deterministic strategies for the k-CTP was fully treated by Westphal [10], families of graphs for which a competitive ratio better than $2k + 1$ can be achieved, remained to be identified. In this context, we designed DETOUR strategy to improve significantly the competitive ratio obtained on graphs satisfying $\mu_{\max} < k$. Its competitive ratio is $2\mu_{\max} + \sqrt{2}(k - \mu_{\max}) + 1$.

Some open questions emerge from this study. First, we wonder if a better strategy exists when $\mu_{\max} < k$. In other words, we do not know whether DETOUR is optimal for this family of graphs. Second, randomized strategies may offer the opportunity to decrease the ratio obtained with the deterministic framework, *i.e.* to go below the slope $\sqrt{2}$ established in Corollary 1. More generally, lots of strategies with a ratio less than $2k + 1$ on certain families of graphs may exist. We believe that local assessments of the competitive ratio can lead us to defeat strategy REPOSITION on many kinds of instances.

References

1. Bar-Noy, A., Schieber, B.: The Canadian Traveller Problem. In: Proc. of ACM/SIAM SODA. pp. 261–270 (1991)
2. Bender, M., Westphal, S.: An optimal randomized online algorithm for the k-Canadian Traveller Problem on node-disjoint paths. J. Comb. Optim. **30**(1), 87–96 (2015)
3. Bergé, P., Hemery, J., Rimmel, A., Tomasik, J.: On the competitiveness of memoryless strategies for the k-Canadian Traveller Problem. In: Proc. of COCOA. pp. 566–576 (2018)
4. Borodin, A., El-Yaniv, R.: Online computation and competitive analysis. Cambridge Univ. Press (1998)
5. Chaourar, B.: A Linear Time Algorithm for a Variant of the MAX CUT Problem in Series Parallel Graphs. Adv. Operations Research (2017)
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1**(1), 269–271 (1959)
7. Haglin, D.J., Venkatesan, S.M.: Approximation and Intractability Results for the Maximum Cut Problem and its Variants. IEEE Trans. Computers **40**(1), 110–113 (1991)
8. Papadimitriou, C., Yannakakis, M.: Shortest paths without a map. Theor. Comput. Sci. **84**(1), 127–150 (1991)
9. Shiri, D., Salman, F.S.: On the randomized online strategies for the k-Canadian traveler problem. J. Comb. Opt. pp. 1–14 (2019)
10. Westphal, S.: A note on the k-Canadian Traveller Problem. Inform. Proces. Lett. **106**(3), 87–89 (2008)
11. Xu, Y., Hu, M., Su, B., Zhu, B., Zhu, Z.: The Canadian traveller problem and its competitive analysis. J. Comb. Optim. **18**(2), 195–205 (2009)