On the competitiveness of memoryless strategies for the Canadian Traveller Problem

Pierre Bergé[a,*], Julien Hemery[b], Arpad Rimmel[b], Joanna Tomasik[b]

[a] *LRI, Université Paris-Sud, Université Paris-Saclay, 91405 Orsay Cedex, France*
[b] *LRI, CentraleSupélec, Université Paris-Saclay, 91405 Orsay Cedex, France*

Abstract

The k-Canadian Traveller Problem, defined and proven PSPACE-complete by Papadimitriou and Yannakakis, is a generalization of the Shortest Path Problem which admits blocked edges. Its objective is to determine the strategy that makes the traveller traverse graph G between two given nodes s and t with the minimal distance, knowing that at most k edges are blocked. The traveller discovers that an edge is blocked when arriving at its endpoint. Westphal showed that the competitive ratio, which is an indicator of the online algorithm quality, of any randomized strategy is not less than $k+1$.

We study the competitiveness of randomized memoryless strategies for the k-CTP. In this context, a decision taken by the traveler in node v of G does not depend on its anterior moves. We establish that the competitive ratio of any randomized memoryless strategy cannot be better than $2k + O(1)$. The primordial consequence of this result is that randomized memoryless strategies are asymptotically as competitive as deterministic strategies which achieve a ratio $2k+1$ at best. In future research, if we aim at designing a strategy with competitive ratio $o(2k)$, we shall focus on strategies which not only are randomized but also use memory.

1. Introduction

The *Canadian Traveller Problem* (CTP), a generalization of the *Shortest Path Problem*, was introduced in [6]. Given an undirected weighted graph $G = (V, E, \omega)$ and two nodes $s, t \in V$, the objective is to design a strategy to make a traveller walk from s to t through G on the shortest path possible. Its particularity is that some edges of G are potentially blocked. The traveller does not know, however, which edges are blocked. This implies that we solve the CTP with online algorithms, called strategies. He discovers blocked edges, also called blockages, when arriving to its endpoints. The k-*Canadian Traveller Problem* (k-CTP) is the parameterized variant of CTP, where an upper

*Corresponding author
Email address: Pierre.Berge@lri.fr (Pierre Bergé)

bound k for the number of blocked edges is given. Both CTP and k-CTP are PSPACE-complete [2, 6].

1.1. State-of-the-art

Strategies for the k-CTP are studied through the competitive analysis, which evaluates their quality [4]. The competitive ratio of a strategy is the maximum, over every satisfiable instance, of the ratio of the distance traversed by the traveller following the strategy and the *optimal offline cost*, which is the distance he traverses if he knows blocked edges from the beginning.

There are two classes of strategies: deterministic and randomized. Regarding the deterministic strategies, Westphal [7] proved that there is no algorithm that achieves a competitive ratio better than $2k + 1$. This ratio is reached by REPOSITION and COMPARISON strategies [7, 8]. The REPOSITION strategy consists in traversing the shortest (s, t)-path on the current graph. If the traveller discovers that an edge e^* of this path is blocked, he goes back to node s and restarts the process on graph G deprived of the edge e^*. This algorithm is executed in polynomial time. However, considering specific practical cases such as urban networks, returning to node s every time the traveller is blocked does not seem realistic. This is why Xu *et al.* [8] introduced the GREEDY algorithm. For grids, it achieves an $\mathcal{O}(1)$ ratio, regardless of k. However, for any graph, this ratio is $\mathcal{O}\left(2^k\right)$.

We evaluate the competitiveness of the randomized strategies by calculating the maximal ratio of the mean distance traversed by the traveller following the strategy by the optimal offline cost. Westphal [7] proved that there is no randomized algorithm that can attain a ratio smaller than $k + 1$. However, unlike the deterministic case, no $(k + 1)$-competitive randomized strategy was identified, excepted for very particular cases. Recently, two randomized algorithms have been proposed. Demaine *et al.* [5] designed a strategy with a ratio $\left(1 + \frac{\sqrt{2}}{2}\right) k + 1$, executed in time of $\mathcal{O}\left(k\mu^2 \left|E\right|^2\right)$ where μ is a parameter which may be exponential. It is dedicated to graphs that can be transformed into apex trees. Bender *et al.* studied in [3] a restriction of k-CTP for graphs composed of node-disjoint (s, t)-paths and proposed a polynomial-time strategy with ratio $(k + 1)$.

1.2. Contributions and paper plan

We study the competitiveness of memoryless strategies [1, 4] for the k-CTP. The choice the traveller makes at node v (to decide where to go next) is independent of his travel before reaching node v. Given that deterministic memoryless strategies cannot achieve a competitive ratio better than $2k+1$, our goal is to prove that randomized memoryless strategies are not more competitive asymptotically. In other words, we aim at identifying a sequence c_k such that the competitiveness of randomized memoryless strategies is larger than c_k with $c_k = 2k + O(1)$.

We remind, in Section 2, the definitions of k-CTP, memoryless strategies and the competitive ratio. In Section 3, we present a set $\mathcal{R}$ (called a road atlas) of *road maps*, *i.e.* pairs (G, E_*) with graph G and blocked edges E_*, on which we study the competitiveness of memoryless strategies. We associate, to any of these road maps, a binary tree representation which allows to understand the behavior of memoryless strategies on these road maps more comfortably. We prove in Section 4 that randomized memoryless strategies cannot drop below a ratio $c_k = 2k + O(1)$ on road maps in $\mathcal{R}$. We also clarify expression $O(1)$ in order to specify the asymptotic behavior of sequence c_k. Eventually, we draw conclusions and highlight the future work in Section 5.

2. Definitions

We start by introducing the notation. For any graph $G = (V, E, \omega)$, let $G \backslash E'$ denotes its subgraph $(V, E \backslash E', \omega)$. If P is a path, we note its cost as $\omega(P) = \sum_{e \in P} \omega(e)$.

2.1. Memoryless Strategies for the k-CTP

We remind the definition of CTP. Let $G = (V, E, \omega)$ be an undirected graph with positive weights. The objective is to make a traveller traverse the graph from a source node s to a target one t, with $s, t \in V$. There is a set $E_* \subsetneq E$ of blocked edges. The traveller does not know a priori which edges are blocked. He discovers a blocked edge only when arriving to one of its endpoints. For example, if (v, w) is a blockage he will discover it when arriving to v (or w). The goal is to design the strategy A with the minimal competitive ratio.

We focus on *memoryless strategies* (MS). Concretely, we suppose that the traveller remembers the blocked edges he has discovered but forgets the nodes which he has already visited. In other words, a decision of an MS is independent of the nodes already visited. In the literature, the term *memoryless* was used in the context of online algorithms (e.g. PAGING PROBLEM [4], LIST UPDATE PROBLEM [1]) which take decisions according to the current state, ignoring past events. An MS can be either deterministic or randomized.

Definition 1 (Memoryless Strategies for the k-CTP). *A deterministic strategy A is an MS if and only if (iff) the next node w the traveller visits depends on graph G deprived of blocked edges already discovered E'_* and the current traveller position v: $w = A(G \backslash E'_*, v)$. Similarly, a randomized strategy A is an MS iff node w is the realization of a discrete random variable $X = A(G \backslash E'_*, v)$.*

MSes are easy to be implemented because they do not use past moves to take a decision. For the same reason, they do not need any extra memory either.

For example, the GREEDY strategy [8] is a deterministic MS. It consists in choosing at each step the first edge of the shortest path between the current node v and the target t. In contrast, the REPOSITION strategy [7] is not an MS as any decision refers to the past moves of the traveller.

We propose the following process to identify whether a strategy A is a deterministic MS. Let us suppose that a traveller T_1 executes strategy A: he has already visited certain nodes of the graph, he is currently at node v but he has not reached target t yet. Let us imagine a second traveller T_2 who is airdropped on node v and starts applying strategy A. If the traveller T_2 always follows the same path as T_1 until reaching t, A is a deterministic MS. If T_1 and T_2 may follow different paths, then A is not an MS. Formally, prooving that a strategy is a MS consists in finding the function which transforms the pair $(G \backslash E_*, v)$ into node $w = A\left(G \backslash E'_*, v\right)$.

2.2. Competitive ratio

Let (G, E_*) be a *road map*, *i.e.* a pair with graph $G = (V, E, \omega)$ and blocked edges $E_* \subsetneq E$, such that there is an (s,t)-path in graph $G \backslash E_*$ (nodes s and t remain in the same connected component when all blocked edges are discovered). We note $\omega_A(G, E_*)$ the distance traversed by the traveller reaching t with strategy A on graph G with blocked edges E_* and $\omega_{\min}(G, E_*)$ the cost of the shortest (s,t)-path in graph $G \backslash E_*$.

The ratio $\omega_A(G, E_*) / \omega_{\min}(G, E_*)$ is abbreviated as $c_A(G, E_*)$. A strategy A is c_A-competitive [4, 8] iff for any $(G, E_*), \omega_A(G, E_*) \leq c_A \omega_{\min}(G, E_*)$. Otherwise stated, for any (G, E_*), $c_A(G, E_*) \leq c_A$. If strategy A is randomized, $\omega_A(G, E_*)$ is replaced by $\mathbb{E}\left(\omega_A(G, E_*)\right)$ which is the expected distance traversed by the traveller to reach t with strategy A. The competitive ratio can also be evaluated on a family $\mathcal{R}$ of road maps, put formally,

$$c_{A,\mathcal{R}} = \max_{(G,E_*)\in\mathcal{R}} c_A(G, E_*) \tag{1}$$

This "local" competitive ratio fulfils $c_{A,\mathcal{R}} \leq c_A$. The definition of the competitive ratio can also be extended to families of strategies. We note c_{MS} the competitive ratio of MSes, which is the minimum over competitive ratios of any MSes: $c_{\text{MS}} = \min_{A\ \text{MS}} c_A$. In the remainder, we identify a set of road maps $\mathcal{R}$ such that the competitive ratio of MSes on it is $2k + O(1)$.

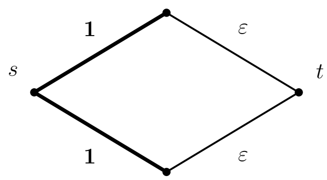
3. Road atlas used to study randomized MSes

We specify a *road atlas* $\mathcal{R}$ which is a set of road maps. To do so, we start by introducing a family of graphs G_k and establishing properties on them. The objective is to use this road altas to assess the competitive ratio of randomized MSes on it.

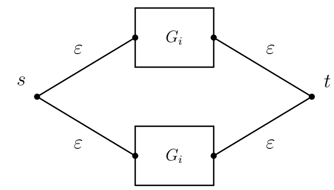
3.1. Graphs G_k

We define recursively a sequence of graphs G_i for $i \geq 1$ with weights from $\{1, \varepsilon\}$, $0 < \varepsilon \ll 1$. Graphs G_1 and G_{i+1} are represented in Figures 1a and 1b, respectively. Graphs G_2 and G_3 are shown in Figure 1c and 1d. Edges with weight 1 are thicker than edges with weight ε. On any graph G_i, axis Δ_{vert} is the vertical axis of symmetry (see Figures 1c and 1d).

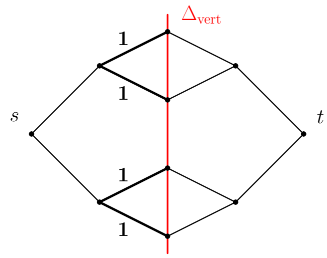
When the traveller discover blocked edges in a graph G, parts of the graph become *dead ends*. If a traveller visits a dead end, the only chance for him to reach t is to leave it anyway. Dead ends only provide to the traveller additional



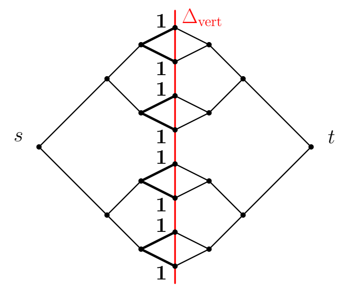
(a) Graph G_1



(b) Graph G_{i+1}



(c) Graph G_2 and axis Δ_{vert}. Weights ε are not indicated to keep the figure readable.

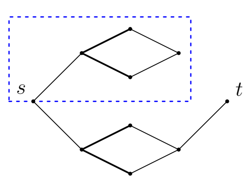


(d) Graph G_3 and axis Δ_{vert}.

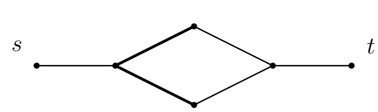
Figure 1: Recursive construction of graphs G_i

costs which make the competitive ratio increase. As a consequence, the most competitive strategies (memoryless or not) do not visit dead ends. From now on, as soon as a dead end appear when the traveller discovers a blockage, we delete it from the graph. Graph $G \backslash E'$ becomes graph G deprived of both edges E' and dead ends.

We focus on road maps (G_k, E_*) composed of graph G_k and k blocked edges E_* at the right of axis Δ_{vert} in graph G_k. Indeed, blocking edges left to Δ_{vert} affects negligibly the total distance traversed by a traveller. Let us suppose that the traveller, guided by a MS, traverses graph G_k and has already discovered some blocked edges $E' \subseteq E_*$. He tries now to reach t on graph $G_k \backslash E'$, ignoring that the undiscovered blocked edges are $E'_* = E_* \backslash E'$. We study the competi-



(a) Graph G_2 deprived of one edge makes appear
a dead end (subgraph in the blue dashed box)



(b) The graph obtained after deleting all nodes/edges forming a dead end

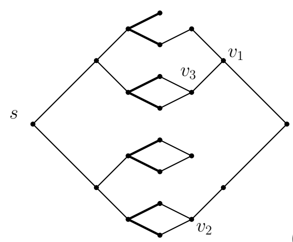
Figure 2: An illustration of dead ends on graph G_2.

tiveness of MSes on the road atlas $\mathcal{R}$ composed of road maps with sub-graphs $G_k \backslash E' \in \mathcal{G}_k$ where $\mathcal{G}_k = \{G_k \backslash E' : |E'| \leq k\}$ and blocked edges E'_* which fulfil $|E'| + |E'_*| = k$. Formally,

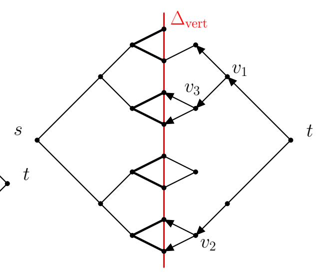
$$\mathcal{R} = \bigcup_{k=1}^{+\infty} \mathcal{R}_k \text{ with } \mathcal{R}_k = \{(G_k \backslash E', E'_*) : G_k \backslash E' \in \mathcal{G}_k, E' \cap E'_* = \emptyset, |E'| + |E'_*| = k\} \tag{2}$$

3.2. Equivalent binary trees

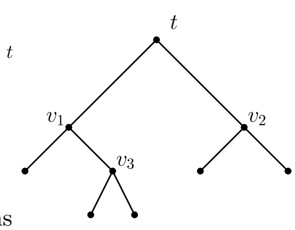
For any graph G of $\mathcal{G}_k$, we define an equivalent binary tree (EBT) noted T_G which represents all changes of direction from t in graph G. We note $T_\varnothing$ the empty tree. Any nonempty tree is a triplet (v, T_a, T_b) with a root $v \in V$ and two subtrees T_a and T_b.



(a) Graph $G \in \mathcal{G}_3$



(b) Changes of directions from t



(c) The EBT T_G

Figure 3: An example of graph $G \in \mathcal{G}_3$ and its EBT T_G.

The EBT is obtained thanks to a breadth-first search (BFS) from t in direction of nodes on axis Δ_{vert}. Node t is the root of the EBT. Graphically, each node from which there are two possible directions for the next move of the BFS provides two extra branches in the tree (see nodes t, v_1, v_2 and v_3 in Figure 3b). Figures 3a, 3b and 3c illustrate the construction of the EBT. To put the definition of EBTs formally, let $\text{BFS}(v)$ denote the set of nodes visited just after node v in the BFS starting from t. For example, $v_1 \in \text{BFS}(t)$ and $v_3 \in \text{BFS}(v_1)$. We force this BFS to stop on axis Δ_{vert} such that $v \in \Delta_{\text{vert}} \Rightarrow \text{BFS}(v) = \emptyset$. We define below function BIN-TREE which gives the construction of tree T_G fulfilling $\text{BIN-TREE}(t) = T_G$.

$$\text{BIN-TREE}(v) = \begin{cases} T_\varnothing \text{ if } \text{BFS}(v) = \emptyset \\ \text{BIN-TREE}(v_{\text{next}}) \text{ if } \text{BFS}(v) = \{v_{\text{next}}\} \\ (v, \text{BIN-TREE}(v_{\text{up}}), \text{BIN-TREE}(v_{\text{down}})) \text{ if } \text{BFS}(v) = \{v_{\text{up}}, v_{\text{down}}\} \end{cases}$$

We introduce notations which characterize some elements in an EBT. For any edge e in EBT T, we note $P(e)$ the parent edge of e. Let $U(e)$ denote the "uncle" of edge e, which has the same parent than $P(e)$. Put formally, $U(e) = \{e' \in T : P(e') = P(P(e)) = P^2(e)\}$. For any EBT T, we define recursively a set of edges $E_T(k)$ which is the set of edges with depth k. Set $E_T(0)$ represents the edges connected to the root t and $E_T(k) = \{e : P(e) \in E_T(k-1)\}$. We say that, for admissible integer D, set $\mathcal{C}_{\mathrm{B}}(D)$ denotes binary trees T which are complete from depth 0 to depth D. In other words, for any $0 \leq j \leq D$, $|E_T(j)| = 2^{j+1}$. The graph given in Figure 4b belongs to $\mathcal{C}_{\mathrm{B}}(1)$.

We pursue our reasoning by establishing properties on EBTs T_G to understand the behavior of MSes on graphs $G \in \mathcal{G}_k$. Theorem 1 states that for any graph $G_k \backslash E' \in \mathcal{G}_k$, its EBT is a complete binary tree from depth 0 to $k-1-|E'|$. It is proven by induction. Figures 4a, 4b and 4c illustrate the induction step on an example of $\mathcal{G}_3$.

Theorem 1. *For any graph $G_k \backslash E' \in \mathcal{G}_k$, $T_{G_k \backslash E'} \in \mathcal{C}_B(k-1-|E'|)$.*

Proof. If $E' = \emptyset$, the EBT T_{G_k} is a complete binary tree, so for any $0 \leq j \leq k-1$, $\left|E_{T_{G_k}}(j-1)\right| = 2^j$. Let $G_k \backslash E'$ be in $\mathcal{G}_k$ with $|E'| = k - D$ and fixed integer $0 < D \leq k-1$. As an induction hypothesis, we assume that, for any $G_k \backslash E'' \in \mathcal{G}_k$ with $|E''| = k-1-D$, $T_{G_k \backslash E''} \in \mathcal{C}_{\mathrm{B}}(D)$. The objective is to prove that $T_{G_k \backslash E'} \in \mathcal{C}_{\mathrm{B}}(D-1)$ with $D = k - |E'|$.

Let e be an arbitrary edge of E'. We note $E' = \{e\} \cup E''$ with $|E''| = k-D-1$. By hypothesis, graph $G_k \backslash E'' \in \mathcal{G}_k$ fulfils $T_{G_k \backslash E''} \in \mathcal{C}_{\mathrm{B}}(D)$. The EBT $T_{G_k \backslash E''}$ contains edge e. We note v the "root" of edge e, *i.e.* node v such that $e = (u, v)$ and $v = P(u)$. Let T_v be the subtree of $T_{G_k \backslash E''}$ with root v and w be the "sibling" of u (said differently the second son of node v). When edge e is removed from $G_k \backslash E''$, not only edge e but also its descendants disappear in the EBT and T_v becomes T_w, the subtree of root w.

Let $T_{G_k \backslash E''} = (t, T_a, T_b)$: we assume w.l.o.g. that $e \in T_a$. We know now that $T_{G_k \backslash E'} = (t, T'_a, T_b)$ where T'_a is tree T_a after replacing subtree T_v by T_w. As $T_{G_k \backslash E''}$ is complete until depth D, T_a and T_b are necessarily complete until edges of depth D. When T_v is replaced by T_w, the global depth of T_a decreases by 1, so T'_a is complete until depth $D-1$. Eventually, the EBT $T_{G_k \backslash E'} = (t, T'_a, T_b)$ is a binary complete tree from depth 0 to depth $D-1$. ■

4. Competitiveness of randomized MSes

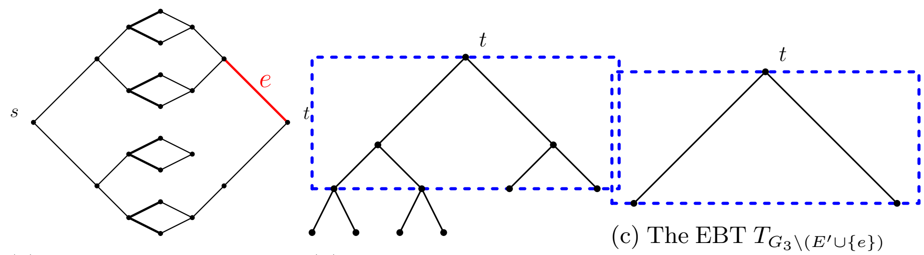
5. Conclusion and further work

References

[1] Susanne Albers. Online algorithms: a survey. *Math. Program.*, pages 3–26, 2003.

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

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



(a) Example of graph $G_3 \backslash E' \in \mathcal{G}_3$, with $|E'| = 1$. Edge e (red) is about to be removed from this graph.

(b) The EBT $T_{G_3 \backslash E'}$ which belongs to $\mathcal{C}_{\mathrm{B}}(1)$ because $k - 1 - |E'| = 1$. Edges of depth 0 and 1 are framed.

(c) The EBT $T_{G_3 \backslash (E' \cup \{e\})}$ which belongs to $\mathcal{C}_{\mathrm{B}}(0)$ because $k - 1 - |E' \cup \{e\}| = 0$. Edges of depth 0 are framed.

Figure 4: Illustrating the proof of Theorem 1 on a graph of $\mathcal{G}_3$.

[4] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

[5] Erik D. Demaine, Yamming Huang, Chung-Shou Liao, and Kunihiko Sadakane. Canadians Should Travel Randomly. *Proc. of ICALP*, pages 380–391, 2014.

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

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

[8] Yinfeng Xu, Maolin Hu, Bing Su, Binhai Zhu, and Zhijun Zhu. The Canadian traveller problem and its competitive analysis. *J. Comb. Optim.*, 18(2):195–205, 2009.