Canadian Travellers should use memory

Pierre Bergé *,[1], Julien Hemery[2], 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, 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 [**?**]. 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 bound k for the number of blocked edges is given. Both CTP and k-CTP are PSPACE-complete [**?**, **?**].

1.1 State-of-the-art

Strategies for the k-CTP are studied through the competitive analysis, which evaluates their quality [**?**]. 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.

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

There are two classes of strategies: deterministic and randomized. Regarding the deterministic strategies, Westphal [?] proved that there is no algorithm that achieves a competitive ratio better than $2k+1$. This ratio is reached by REPOSITION and COMPARISON strategies [?, ?]. 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.* [?] 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 [?] 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.* [?] 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 [?] 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 [?, ?] 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 [**?**], LIST UPDATE PROBLEM [**?**]) 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 [**?**] 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 [**?**] 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 $w = A(G \backslash E'_*, v)$.

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 [**?**, **?**] 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 $\mathbb{E}(\omega_A(G, E_*))$ 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,

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

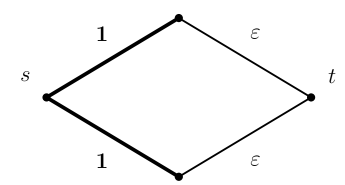
This "local" competitive ratio fulfils $c_{A,\mathcal{R}} \leq c_A$. The definition of the competitive ratio can also be extended to families of strategies. We note c_{MS} the competitive ratio of MSes, which is the minimum over competitive ratios of any MSes: $c_{\mathrm{MS}} = \min_{A \text{ MS}} c_A$. 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

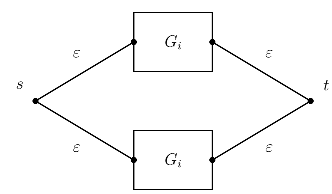
We specify a *road atlas* $\mathcal{R}$ which is a set of road maps. To do so, we start by introducing a family of graphs G_k and establishing properties on them. The objective is to use this road altas to assess the competitive ratio of randomized MSes on it.

3.1 Graphs G_k

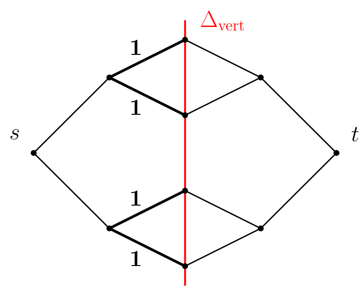
We define recursively a sequence of graphs G_i for $i \geq 1$ with weights from $\{1, \varepsilon\}$, $0 < \varepsilon \ll 1$. Graphs G_1 and G_{i+1} are represented in Figures 1a and 1b, respectively. Graphs G_2 and G_3 are shown in Figure 1c and 1d. Edges with weight 1 are thicker than edges with weight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



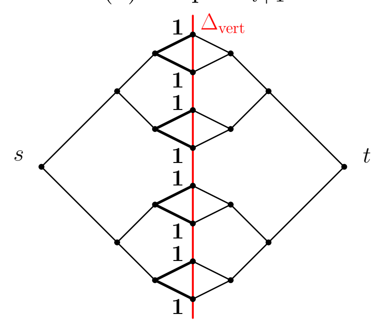
(a) Graph G_1



(b) Graph G_{i+1}



(c) Graph G_2 and axis Δ_{vert}. Only edges right to this axis can be blocked.



(d) Graph G_3 and axis Δ_{vert}. Weights ε are not indicated to keep the readability.

Figure 1: Recursive construction of graphs G_i

When the traveller discover blocked edges in G_k, 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 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. Let $\mathcal{G}_k$ denote the set of all sub-graphs of G_k where at most k edges on the right have been removed. Formally, $\mathcal{G}_k = \{G_k \backslash E' : |E'| \leq k\}$. The road atlas $\mathcal{R}$ is composed of road maps with sub-graphs $G_k \backslash E' \in \mathcal{G}_k$ and blocked edges E_* which fulfil $|E'| + |E_*| = k$.

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

3.2 Equivalent binary trees

We define for each graph of $\mathcal{G}_k$ an equivalent binary tree, which will represent all changes of direction from t. We note $T_\varnothing$ the empty tree, and (v, T_a, T_b) a non empty tree with $v \in V$, T_a the binary tree above and T_b the binary tree below.

We obtain the equivalent binary tree by applying the BIN-TREE algorithm. This algorithm take as an entry a directed tree obtained by first removing the left part of the graph and then directed all edges from t. Then, the BIN-TREE algorithm will construct a binary tree according to the outdegree of each node.

Algorithm 1: BIN-TREE algorithm

Data: a directed tree T and a node v
Result: binary tree
if $deg^-(v) = 0$ **then** **return** $T_\varnothing$;
if $deg^-(v) = 1$ **then** **return** BIN-TREE(T, v_{next}) ;
if $deg^-(v) = 2$ **then** **return** (v, BIN-TREE(T, v_{up}), BIN-TREE(T, v_{down}) ;

Given an edge in a binary tree, we will note $A(e)$ the child edge from above and $B(e)$ the child edge from below. We also note $C(e)$ the set of all the descendants of e. Formally, $C(e) = C(A(e)) \cup C(B(e)) \cup \{A(e), B(e)\}$. Finally, we note $P(e)$ the parent edge of e : $P(e) = e' \Leftrightarrow e = A(e')$ or $e = B(e')$.

We define recursively a set of edges E_k, E_0 being the edges connected to the root and $E_k = \{e : P(e) \in E_{k-1}\}$. E_k represents the edges which are at distance k from the root.

When an edge is cut from the original graph, the equivalent tree can be easily obtain by replacing the tree (v, T_a, T_b) by T_a (resp. T_b) if the cut edge is before T_b (resp. T_a)

Lemma 1 *For any road map $(G, E_*) \in \mathcal{R}_k$, $E_{T_G}(j-1) = 2^j$ with $j = |E_*|$*

Proof. If $G = G_k$, then $|E_*| = k$ and the binary tree is complete, so $E_{T_{G_k}}(k-1) = 2^k$.

We suppose now for any road map with $|E_*| = j$ that $E_{T_G}(j-1) = 2^j$. Let (G, E_*) a road map with $|E_*| = j - 1$ Then it exists a road map $(G', E_* \cup \{e\})$ with G obtained by removing e to G'.

First, we remark that for any binary tree $T = (v, T_a, T_b)$ and i, if the number of edge of depth i is 2^{i+1}, the tree is complete until depth i, so T_a and T_b are complete until depth $i - 1$ and each have 2^i edges at depth $i - 1$: $|E_T(i)| = 2^{i+1} \implies |E_{T_a}(i-1)| = |E_{T_b}(i-1)| = 2^i$.

Second, we remark that for any binary tree T and i, if the number of edge of depth i is 2^{i+1}, then the number of edge for the depth $i - 1$ is 2^i : $|E_T(i)| = 2^{i+1} \implies |E_T(i-1)| = 2^i$.

Let $T' = (v, T_a, T_b)$ the sub-tree of T'_G where v is the root of the edge e (we will suppose e is on the T_a side). Then if e is at depth i then the remarks and the hypothesis give $|E_T(j-i-1)| = 2^{j-i}$ and $|E_T(j-i-2)| = 2^{j-i-1} = |E_{T_a}(j-i-2)|$. By cutting e, T become in G the tree T_a. We find that the number of edge at the depth $j - i - 2$ remains. So the rest of the tree being unchanged by cutting e, the number of edges of T_G for depth $j - 2$ is unchanged from T'_G and is equal to 2^{j-1} : $|E_{T_G}(j-2)| = 2^{j-1}$. ■

4 Competitiveness of randomized MSes

5 Conclusion and further work

References