The Canadian Traveller Problem on Outerplanar Graphs

Laurent Beaudou
Université Clermont-Auvergne, CNRS, Mines de Saint-Etienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France

Pierre Bergé
Université Clermont-Auvergne, CNRS, Mines de Saint-Etienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France

Vsevolod Chernyshev
Université Clermont-Auvergne, CNRS, Mines de Saint-Etienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France

Antoine Dailly
Université Clermont-Auvergne, CNRS, Mines de Saint-Etienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France

Yan Gerard
Université Clermont-Auvergne, CNRS, Mines de Saint-Etienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France

Aurélie Lagoutte
Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France

Vincent Limouzy
Université Clermont-Auvergne, CNRS, Mines de Saint-Etienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France

Lucas Pastor
Université Clermont-Auvergne, CNRS, Mines de Saint-Etienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France

Abstract

We study the k-Canadian Traveller Problem, where a weighted graph $G = (V, E, \omega)$ with a source $s \in V$ and a target $t \in V$ are given. This problem also has a hidden input $E_* \subsetneq E$ of cardinality at most k representing blocked edges. The objective is to travel from s to t with the minimum distance. At the beginning of the walk, the blockages E_* are unknown: the traveller discovers that an edge is blocked when visiting one of its endpoints. Online algorithms, also called strategies, have been proposed for this problem and assessed with the competitive ratio, *i.e.*, the ratio between the distance actually traversed by the traveller divided by the distance he would have traversed knowing the blockages in advance.

Even though the optimal competitive ratio is $2k + 1$ even on unit-weighted planar graphs of treewidth 2, we design a polynomial-time strategy achieving competitive ratio 9 on unit-weighted outerplanar graphs. This value 9 also stands as a lower bound for this family of graphs as we prove that, for any $\varepsilon > 0$, no strategy can achieve a competitive ratio $9 - \varepsilon$. Finally, we show that it is not possible to achieve a constant competitive ratio (independent of G and k) on weighted outerplanar graphs.

2012 ACM Subject Classification Theory of computation → Online algorithms; Theory of computation → Graph algorithms analysis; Mathematics of computing → Graph algorithms

Keywords and phrases Canadian Traveller Problem, Online algorithms, Competitive analysis, Outerplanar graphs

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.19

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 19; pp. 19:1–19:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version *Full Version*: https://arxiv.org/abs/2403.01872 [6]

Funding This work was supported by the International Research Center “Innovation Transportation and Production Systems” of the I-SITE CAP 20-25 and by the ANR project GRALMECO (ANR-21-CE48-0004).

1 Introduction

The k-*Canadian Traveller Problem* (k-CTP) was introduced by Papadimitriou and Yannakakis [24]. It models the travel through a graph where some obstacles may appear. Given an undirected weighted graph $G = (V, E, \omega)$, with $\omega : E \to \mathbb{Q}^+$, and two of its vertices $s, t \in V$, a traveller walks from s to t on G despite the existence of blocked edges $E_* \subsetneq E$ (also called *blockages*), trying to contain the length of his walk. The traveller does not know which edges are blocked when he begins his journey. He discovers that an edge $e = uv$ is blocked, *i.e.*, belongs to E_*, when he visits one of its endpoints u or v. The parameter k is an upper bound on the number of blocked edges: $|E_*| \leq k$. Several variants have also been studied: where edges are blocked with a certain probability [1, 5, 13, 19], with multiple travellers [11, 25], where we can pay to sense remote edges [19], or where we seek the shortest tour [20, 22]. This problem has applications in robot routing for various kinds of logistics [1, 2, 8, 18, 23].

For a given walk on the graph, its *cost* (also called distance) is the sum of the weights of the traversed edges. The objective is to minimize the cost of the walk used by the traveller to go from s to t. A pair (G, E_*) is called a *road map*. All the road maps considered are feasible: there exists an (s, t)-path in $G \setminus E_*$, the graph G deprived of E_*. In other words, there is always a way to reach target t from source s despite the blockages.

A solution to the k-CTP is an online algorithm, called a *strategy*, which guides the traveller through his walk on the graph : given the input graph, the history of visited nodes, and the information collected so far (here, the set of discovered blocked edges), it tells which neighbor of the current vertex the traveller should visit next. The quality of the strategy can be assessed with competitive analysis [14]. 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 in advance. The k-CTP is PSPACE-complete [5, 24] in its decision version that asks, given a positive number r and the input weighted graph, whether there exists a strategy with competitive ratio at most r. Westphal [26] proved that no deterministic strategy achieves a competitive ratio less than $2k + 1$ on all road maps satisfying $|E_*| \leq k$. 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 $2k + 1$. Randomized strategies have also been studied, see *e.g.* [10, 16].

Our goal is to distinguish between graph classes on which the k-CTP has competitive ratio $2k + 1$ (the optimal ratio for general graphs) and the ones for which this bound can be improved. This direction of research has already been explored in [12]: there is a polynomial-time deterministic strategy which achieves ratio $\sqrt{2}k + O(1)$ on graphs with bounded-size maximum (s, t)-cuts. We pursue this study by focusing on a well-known family of graphs: outerplanar graphs, which are graphs admitting a planar embedding (without edge-crossing) where all the vertices lie on the outer face. In [12], an outcome dedicated to a superclass of weighted outerplanar graphs implies that there is a strategy with ratio $2\frac{3}{4}k + O(1)$ on them. Interestingly, however, even very simple unit-weighted planar graphs of treewidth 2, consisting only of disjoint (s, t)-paths, admit the general ratio $2k + 1$ as optimal [15, 26].

Our results and outline. After some preliminaries (Section 2), we describe in Section 3 a polynomial-time strategy achieving a competitive ratio 9 on instances where the input graph is a unit-weighted outerplanar graph:

▶ **Theorem 1.1.** *There is a strategy with competitive ratio 9 for unit-weighted outerplanar graphs.*

In the input outerplanar graph, vertices s and t lie on the outer face. The latter can be seen (provided 2-connectedness) as a cycle embedded in the plane, allowing to explore two sides when we travel from s to t (the two sides are the two internally disjoint (s,t)-paths forming the cycle). The core of the strategy consists in an exploration of both sides via a so-called *exponential balancing*. Then, the most technical part consists in the handling of the chords linking both sides. We maintain a competitiveness invariant of the strategy which produces a final ratio of 9.

Note that Theorem 1.1 can be extended as a corollary to outerplanar graphs where the *stretch*, defined as the ratio between the maximum and minimum weight, is bounded by some fixed S. In this case, the strategy has ratio $9S$.

Surprisingly, the k-CTP on unit-weighted outerplanar graphs has connections with another online problem called the *linear search* problem [3, 7, 9] or the *cow-path* problem [21]. In this problem, a traveller walks on an infinite line, starting at some arbitrary point, and its goal is to reach some target fixed by the adversary. It was shown that applying an exponential balancing on this problem is the optimal way, from the worst case point of view, to reach the target [3]. We explain in Section 3.3 why, on unit-weighted outerplanar graphs, the competitive ratio stated in Theorem 1.1 is optimal and how it can be deduced from the literature on the linear search problem.

▶ **Theorem 1.2.** *For any $\varepsilon > 0$, no deterministic strategy achieves competitive ratio $9-\varepsilon$ on all road maps (G, E_*), where G is a unit-weighted outerplanar graph.*

Finally, in Section 4, we show that no constant competitive ratio can be achieved on outerplanar graphs where weights can be selected arbitrarily.

▶ **Theorem 1.3.** *There is no constant C, independent from G and k, such that a deterministic strategy achieves competitive ratio C on all road maps (G, E_*) where G is a weighted outerplanar graph.*

We summarize in Table 1 the state-of-the-art of the competitive analysis of deterministic strategies for the k-CTP, giving for each family an upper bound of competitiveness (*i.e.*, a strategy with such ratio exists) and a lower bound (*i.e.*, no strategy can achieve a smaller ratio). Our contributions are framed.

Due to space limitation, the proofs of results marked with **(*)** are omitted here and available in the full version [6].

2 Definitions and first observations

2.1 Graph preliminaries

We work on undirected connected weighted graphs $G = (V, E, \omega)$, where $\omega : E \to \mathbb{Q}^+$. A graph is *equal-weighted* (resp. *unit-weighted*) if the value of $\omega(e)$ is the same (resp. 1) for every edge $e \in E$. This article follows standard graph notations from [17]. We denote by $G[U]$ the subgraph of G induced by $U \subseteq V$: $G[U] = \left(U, E[U], \omega_{|E[U]}\right)$; and by $G \setminus U$ the graph deprived of vertices in U: $G \setminus U = G[V \setminus U]$. A *simple* (u,v)-*path* is a sequence of

Table 1 Deterministic strategies performances for the k-CTP.

Family of graphs	upper bound	lower bound
unit-weighted planar of treewidth 2	$2k+1$ [26]	$2k+1$ [15, 26]
bounded maximum edge (s,t)-cuts	$\sqrt{2}k + O(1)$ [12]	?
outerplanar	$2^{\frac{3}{4}}k + O(1)$ [12]	not constant
unit-weighted outerplanar	9	9

pairwise different vertices between u and v, while, in a (u,v)-*walk*, vertices can be repeated. The *cost* (or *traversed distance*) of a walk or a path is the sum of the weights of the edges it traverses. A vertex v is an *articulation point* if $G \setminus \{v\}$ is not connected.

An (s,t)-*separator* $X \subsetneq V \setminus \{s,t\}$ in graph G is a set of vertices such that s and t are disconnected in graph $G \setminus X$. We denote by $R_G(s,X)$ (resp. $R_G(t,X)$) the *source* (resp. *target*) *component* of separator X, which is a set made up of the vertices of X together with all vertices reachable from s (resp. t) in $G \setminus X$.

A graph is *outerplanar* if it can be embedded in the plane in such a way that all vertices are on the outer face. An outerplanar graph is 2-connected if and only if the outer face forms a cycle. Given an embedding of a 2-connected outerplanar graph $G = (V,E)$ and two vertices s and t, let $s \cdot p_1 \cdot p_2 \cdots p_h \cdot t \cdot q_1 \cdot q_2 \cdots q_\ell \cdot s$ be the cycle along the outer face of G and let $S_1 = \{p_1, p_2, \ldots, p_h\}$ and $S_2 = \{q_1, q_2, \ldots, q_\ell\}$ with $V = \{s,t\} \cup S_1 \cup S_2$. We can slightly deform the embedding so that s and t are aligned along the horizontal axis; since the outer face forms a cycle, we will refer to S_1 (resp. S_2) as the *upper* (resp. *lower*) *side* of G. A chord xy of the cycle formed by the outer face is said to be (s,t)-*vertical* (resp. (s,t)-*horizontal*) if x and y belong to different sides (resp. to the same side), see Figure 1. When $x = s$ or $y = t$, the chord is considered as (s,t)-horizontal and not (s,t)-vertical. Any (s,t)-vertical chord (simply *vertical chord* when the context is clear) is an (s,t)-separator. Considering a set of vertical chords, we say that the *rightmost* one has the minimal inclusion-wise target component. Due to planarity, the rightmost vertical chord is unique for any such set.

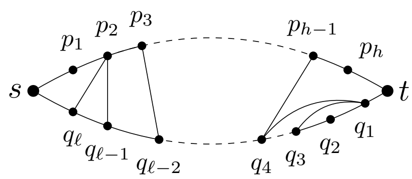

Figure 1 Example of an outerplanar graph: p_2q_ℓ, $p_2q_{\ell-1}$, $p_3q_{\ell-2}$, and $p_{h-1}q_4$ are vertical chords and q_1q_3, q_1q_4 are horizontal chords.

2.2 Problem definition and competitive analysis

Let $G = (V,E,\omega)$ be a graph and E_* represent a set of blocked edges. A pair (G,E_*) is a *road map* if s and t are connected in $G \setminus E_*$.

▶ **Definition 2.1** (k-CTP).

Input: *A graph* $G = (V,E,\omega)$*, two vertices* $s,t \in V$*, and a set* E_* *of blocked edges which are unknown such that* $|E_*| \leq k$ *and* (G,E_*) *is a road map.*

Objective: *Traverse graph* G *from* s *to* t *with minimum cost.*

A solution to the k-CTP is an (s,t)-walk. The set of blocked edges E_* is a hidden input at the beginning of the walk. We say an edge is *revealed* when one of its endpoints has already been visited. A *discovered blocked edge* is a revealed edge which is blocked. At any moment of the walk, we usually denote by $E'_* \subseteq E_*$ the set of discovered blocked edges, in other words the set of blocked edges for which we visited at least one endpoint. During the walk, we are in fact working on $G \setminus E'_*$ as discovered blocked edges can be removed from G.

We call a path *blocked* if one of its edges was discovered blocked; *apparently open* if no blocked edge has been discovered on it for now (it may contain a blocked edge which has not been discovered yet); *open* if we are sure that it does not contain any blocked edge (either all of its edges were revealed open, or it is apparently open and $|E'_*| = k$, or by connectivity considerations since s and t must stay connected in road maps).

For any $F \subseteq E_*$ and two vertices x, y of G, let $d_F\,(G, x, y)$ be the cost of the shortest (x,y)-path in graph $G \setminus F$. If the context is clear, we will use $d_F\,(x, y)$.

We denote by P_{opt} some *optimal offline path* of road map (G, E_*): it is one of the shortest (s,t)-paths in the graph $G \setminus E_*$. Its cost, the *optimal offline cost*, given by $d_{\text{opt}} = d_{E_*}\,(s,t)$, is the distance the traveller would have traversed if he had known the blockages in advance. Given a strategy A for the k-CTP, the *competitive ratio* [14] $c_A(G, E_*)$ over road map (G, E_*) is defined as the ratio between the cost $d_A^{\text{Tr}}\,(G, E_*)$ of the traversed walk and d_{opt}. Formally:

$$c_A(G, E_*) = \frac{d_A^{\text{Tr}}\,(G, E_*)}{d_{\text{opt}}}.$$

Given a monotone family of graphs $\mathcal{F}$ (i.e. closed under taking subgraph), we say that a strategy A admits a competitive ratio $c(k)$ for the family $\mathcal{F}$ if it is an upper bound for all values $c_A\,(G, E_*)$ over all k-CTP road maps (G, E_*) such that $G \in \mathcal{F}$. Conversely, we say that some ratio $c(k)$ cannot be achieved for family $\mathcal{F}$ if, for every strategy A, there is a road map (G, E_*) with $G \in \mathcal{F}$ such that $c_A(G, E_*) > c(k)$.

Westphal [26] identified, for any integer k, a relatively trivial family of graphs for which any deterministic strategy achieves ratio at least $2k + 1$. These graphs are made up of only $k + 1$ identical disjoint (s,t)-paths: they are planar and have treewidth 2. As those paths are indistinguishable, the traveller might have to traverse k of them before finding the open one. This outcome still works if we restrict ourselves to unit weights [15]. Conversely, there are two strategies in the literature achieving competitive ratio $2k + 1$ on general graphs: REPOSITION [26] and COMPARISON [27].

Note that articulation points allow a preliminary decomposition and simplification of any input graph, before even exploring:

▶ **Lemma 2.2.** *Let $\mathcal{F}$ be a monotone family of graphs, and assume that we have a strategy A achieving competitive ratio C on graphs of $\mathcal{F}$ that do not contain any articulation point. Then, there exists a strategy A' achieving the same competitive ratio C on all graphs of $\mathcal{F}$.*

Proof. The strategy A' goes as follows: let (G, E_*) be a road map with $G \in \mathcal{F}$. If G does not contain any articulation point, apply strategy A. Otherwise, let z be an articulation point of G. If $\{z\}$ is not an (s,t)-separator, then, recursively apply strategy A' on $R_G(s, \{z\})$, which is both the source and the target component, to reach t from s. Otherwise (so $\{z\}$ is an (s,t)-separator), recursively apply strategy A' on the source component $R_G(s, \{z\})$ to reach z from s, then recursively apply strategy A' on the target component $R_G(t, \{z\})$ to reach t from z. The procedure is illustrated in Figure 2.

We prove by induction on the number p of articulation points that A' terminates and achieves competitive ratio C. The base case $p = 0$ holds by property of A. For the inductive step, we distinguish two cases. If $\{z\}$ is not an (s,t)-separator, the walk we obtain is of

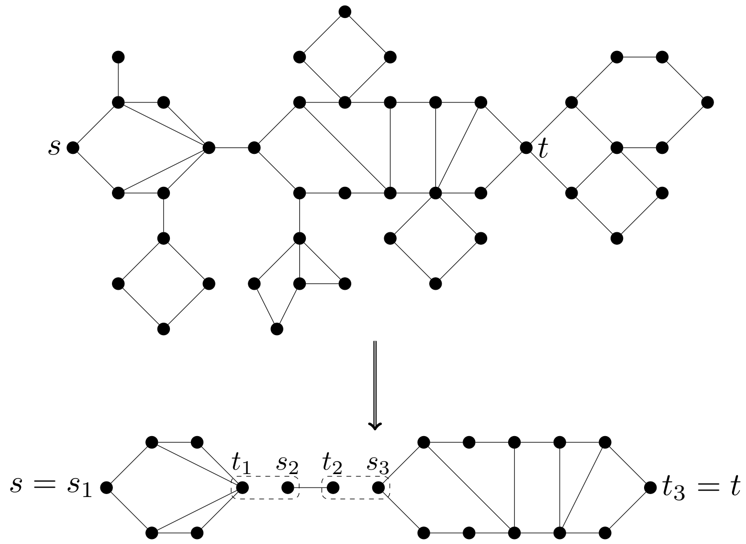

Figure 2 Decomposing the graph into components with no articulation points and removing the useless components (the vertices in a dashed rectangle are the same in the original graph).

length at most Cd_{opt}, which gives competitive ratio C. Otherwise, the length of the whole walk at most $Cd_{E_*}(s,z) + Cd_{E_*}(z,t)$. Since z is an (s,t)-separator, $z \in P_{\text{opt}}$ and we have $d_{\text{opt}} = d_{E_*}(s,z) + d_{E_*}(z,t)$, which concludes the proof. ◀

3 Optimal competitive ratio 9 for unit-weighted outerplanar graphs

We propose a polynomial-time strategy called ExpBalancing dedicated to unit-weighted outerplanar graphs. We show that it achieves competitive ratio 9 for this family of graphs, which we will later prove is optimal (see Theorem 1.2).

3.1 Presentation of the strategy

First, note that Lemma 2.2 allows us to work on outerplanar graphs without articulation points. The input is a unit-weighted 2-connected outerplanar graph G and two vertices s and t. We provide a detailed description of the strategy ExpBalancing that we follow to explore the graph G.

1. **Reaching *t*.** If, at any point in our exploration, we reach t, then we exit the algorithm and return the processed walk.
2. **Horizontal chords treatment.** If, at any point in our exploration, we visit a vertex $u \in S_i$, $i \in \{1,2\}$, incident with an open horizontal chord uv revealed for the first time, then we can remove all the vertices on side S_i that lie between u and v on the outer face. Said differently, we get rid of the vertices which are surrounded by the chord uv. If several horizontal chords incident with u are open, then it suffices to apply this rule to the chord which surrounds all others. This procedure comes from the observation that, due to both unit weights and planarity, the open horizontal chord uv with the rightmost v is necessarily the shortest way to go from u to t on side S_i, and thus visiting the vertices surrounded by it will occur an extra, useless cost.

3. **Exponential balancing.** The core *exponential balancing* principle of the strategy consists in alternately exploring sides within a given budget that doubles each time we switch sides. The budget is initially set to 1. Hence, we walk first on side S_1 with budget 1, second on side S_2 with budget 2, then on side S_1 with budget 4, and so on. We say each budget corresponds to an *attempt*. During each attempt, we traverse a path starting from the source s and stay exclusively on some side S_i, $i \in \{1,2\}$. As evoked in the previous step, at each newly visited vertex, we use an open horizontal chord from our position which brings us as close as possible to t on our side. Either a horizontal chord is open and we use the one which surrounds all other open chords, or if no such chord is open, we pursue our walk on the outer face.
   This balancing process can be described on an automaton depicted in Figure 3 which will be particularly useful in the analysis of this strategy. Here, we assume that we neither are completely blocked on one side nor reveal an open vertical chord. We will handle these cases in Steps 4-6.
   We start our walk on s (state $\mathbf{E_1}$), make an attempt on an arbitrary side (say S_1) with budget 1 (state $\mathbf{E_2}$), and decide to come back to s if t was not reached. During our first attempt on side S_2 with budget 2, we cross a first edge and reach state **A**. Then, we cross a second edge if we are not blocked, but this part of the journey corresponds to the transition between states **A** and **B**. The automaton works as follows:
   - In state **A**, we have explored D vertices on each side (in the description above, $D = 1$ when we first arrive in state **A**). Call x and y the last explored vertices on each side, assume we are on x. The current budget is $2D$ and we pursue our attempt on the side of x.
   - We then explore at most D more vertices on the side of x. We reach state **B**.
   - We then go back to y through s, reaching state **C**.
   - We explore at most D more vertices on the side of y. We go back to state **A** with an updated value of D that is doubled, update x and y, and the sides are switched.
4. **Bypassing a blocked side.** If, during some attempt on side S_i, we are completely blocked (there is no open (s,t)-path on $G[S_i] \setminus E'_*$) before reaching the budget, hence exploring αD ($\alpha < 1$) instead of D (see Figures 4a and 4b), then we backtrack to s and pursue the balancing on the other side S_j ($j \in \{1,2\}, j \neq i$). However, we forget any budget consideration: we travel until we either reach t or visit the endpoint u of some open vertical chord uv. In case there are several open vertical chords incident with u revealed at the same time, we consider the rightmost one. At this moment, we update the current graph $G \setminus E'_*$ by keeping only the target component of separator $\{u,v\}$ and considering u as a new source. Concretely, we concatenate the current walk computed before arriving at u with a recursive call of ExpBalancing on input $(G[R_G(t,\{u,v\})], u, t)$.
5. **Handling open vertical chords between states A and B.** If, during some attempt on side S_i, especially in the transition between states **A** and **B**, we reveal an open vertical chord uv, $u \in S_i$, after having explored distance αD (parameter α is rational, $0 < \alpha \leq 1$, but αD is an integer), then we go to the other side S_j, $j \neq i$, through uv and explore side S_j from v towards s until we:
   - either "see" a vertex y already visited after distance βD (we fix $\beta D \leq \alpha D - 1$, so $0 \leq \beta < \alpha$),
   - or explore distance $\alpha D - 1$ and do not see any already visited vertex,
   - or are completely blocked on S_j before we reach distance $\alpha D - 1$.

   By "see", we mean that we can reach - or not - a neighbor of y which reveals the status of the edge between them: in this way, we actually know the distance to reach y from v even if we did not visit v. Figure 4c describes this rule with an example.

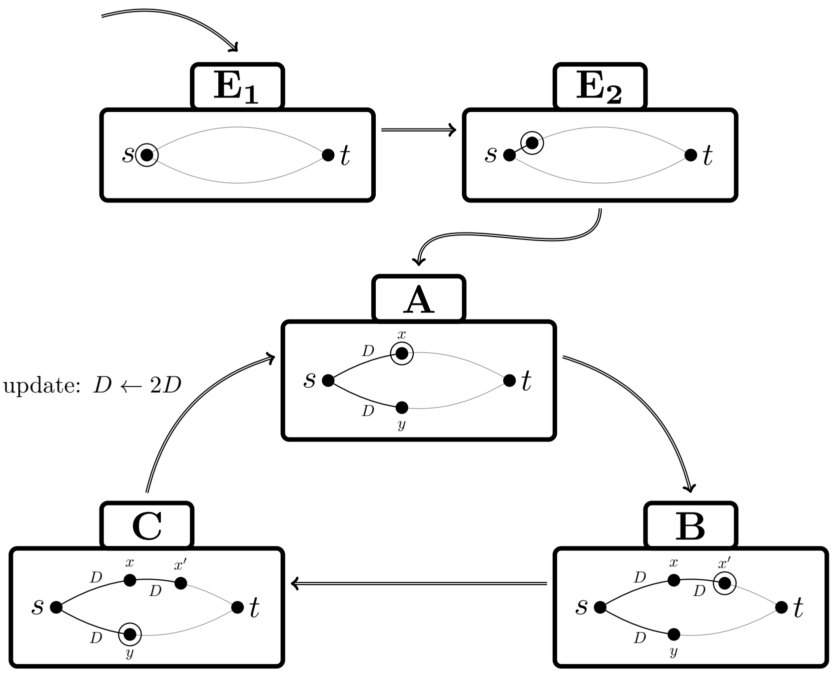

Figure 3 Representation of the exponential balancing divided into three different states. The circled vertex is the one we are currently exploring.

The role of this procedure is to know which endpoint of the chord is the closest to s. If we see, after distance $\beta D = \alpha D - 1$, an already visited vertex (denoted by y in Figure 4c) at distance αD from v, then, we continue the exponential balancing: we go back to v and thus to state **A** in the automaton, update the budget value D which becomes $D + \alpha D$.

Otherwise, we update G by keeping only the target component of separator $\{u, v\}$. The current graph becomes $G' = G[R(t, \{u, v\})]$. If we saw an already visited vertex $y \in S_j$ by exploring distance $\beta D < \alpha D - 1$, then the new source becomes $s' = v$. Otherwise, the new source is $s' = u$. We concatenate the current walk with the walk returned by applying ExpBalancing on input (G', s', t).

6. **Handling open vertical chords between states C and A.** If, during the transition between states **C** and **A** (when some attempt is launched on the side of y and the traversed distance on the other side is larger, see Figure 3), an open vertical chord uv is revealed (see Figure 4d), then we keep only the target component of $\{u, v\}$ and set u as the new source. More formally, we concatenate the current walk with the walk returned by applying ExpBalancing on input (G', u, t), where $G' = G[R_G(t, \{u, v\})]$.

Steps 4–6 can be summarized in this way: when we reveal an open vertical chord uv such that $d_{E_*}(s, v) = d_{E_*}(s, u) + 1$, we launch a recursive call on the target component of separator $\{u, v\}$ with source u and target t. Indeed, any optimal offline path must pass through separator $\{u, v\}$ and, as $d_{E_*}(s, v) = d_{E_*}(s, u) + 1$, we can say there is one optimal offline path P_{opt} such that $u \in P_{\text{opt}}$. Hence, it makes sense to select u as a new source, there is no interest in visiting vertices different from $\{u, v\}$ belonging to their source component.

(a) Step 4 : blocked edge e between states **A** and **B**.

(b) Step 4 : blocked edge e between states **C** and **A**.

(c) Step 5 : open vertical chord uv between states **A** and **B**.

(d) Step 6 : open vertical chord uv between states **C** and **A**.

Figure 4 Four situations potentially met with ExpBalancing on some unit-weighted outerplanar graph.

3.2 Competitive analysis

We now show that the strategy ExpBalancing presented above has competitive ratio 9 on unit-weighted outerplanar graphs. We prove this statement by minimal counterexample. In this subsection, let G denote the smallest (by number of vertices, then number of edges) unit-weighted outerplanar graph on which ExpBalancing does not achieve competitive ratio 9. We will see that the existence of such a graph G necessarily implies a contradiction.

Examples of executions of ExpBalancing are given in Figures 5 and 6.

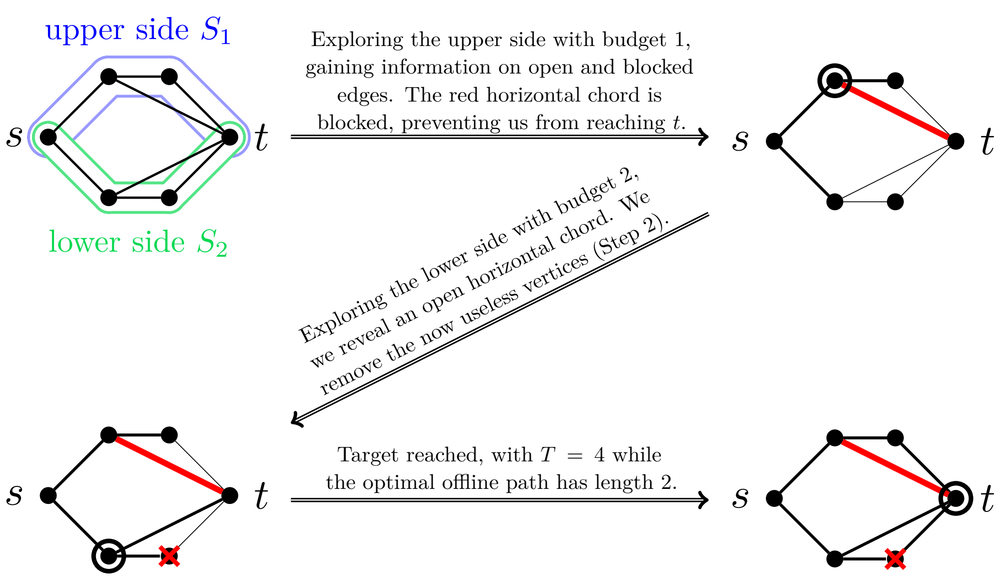

Figure 5 Application of ExpBalancing on the first graph of the decomposition of Figure 2. At each step, the circled vertex is the one we are currently exploring, and we know the status of the bold edges: black is open, red is blocked.

The two following technical lemmas prove that a recursive call has to happen when ExpBalancing is applied on G (Lemma 3.1) and that such a recursive call implies certain properties (Lemma 3.2).

▶ **Lemma 3.1.** *During the execution of* ExpBalancing, *let T be the distance travelled at a given point before the first recursive call (if any). Then, $T \leq 9d_{\text{opt}}$. Moreover, if we are in state $\mathbf{A}$, let x and y be the last two vertices explored on each side during the exponential balancing. Then: (i) $d_{E_*}(s,x) = D$, (ii) $d_{E_*}(s,y) = D$ and (iii) $T \leq 5D$.*

Proof. Assume that we have applied ExpBalancing on G until a certain point and that no recursive call was launched so far. We first focus on the second part of the invariant we want to show:

$$\text{In state } \mathbf{A},\ (i)\ d_{E_*}(s,x) = D,\ (ii)\ d_{E_*}(s,y) = D \text{ and } (iii)\ T \leq 5D.$$

Items (i) and (ii) are true, since no shortcut between s and either x or y can exist: any open horizontal chord is used, and an open vertical chord opening up a shortcut leads to a recursive call (Steps 5 and 6).

Item (iii) is trivially true when we kick-start the exponential balancing: when entering $\mathbf{A}$ from $\mathbf{E_2}$, we have $T = 3$ and $d_{E_*}(s,x) = d_{E_*}(s,y) = 1$. Assume that it is true for a given $D \geq 1$, and let T_0 be the value of T at this point. When we reach state $\mathbf{B}$, we have $T = T_0 + D \leq 6D$. When we reach state $\mathbf{C}$, we have $T = T_0 + D + 3D \leq 9D$. In brief, from state $\mathbf{A}$ to $\mathbf{C}$, we have $d_{\text{opt}} \geq D$ as distance D was explored on both sides without reaching t. The largest ratio of T by D on these phases is 9 at state $\mathbf{C}$, where we have $T \leq 9d_{\text{opt}}$.

During the transition from $\mathbf{C}$ to $\mathbf{A}$, if $D + \alpha D$ denotes the traversed distance on current side at any moment (see Figure 3), then $d_{\text{opt}} \geq D + \alpha D$ and $T = 9D + \alpha D$. The ratio $\frac{T}{d_{\text{opt}}}$ admits a decreasing upper bound, from 9 in state $\mathbf{C}$ to 5 in $\mathbf{A}$. Indeed, when we are back to state $\mathbf{A}$, we have $T = T_0 + D + 3D + D$, but the value of D is updated. Let $D' = 2D$. We have $T = T_0 + 5D \leq 5D + 5D = 5D'$, and so item (iii) remains true during the core loop.

We also have to check that it is true when we met an open vertical chord uv between states $\mathbf{A}$ and $\mathbf{B}$ which satisfies $d_{E_*}(s,v) = d_{E_*}(s,u)$ (case $\beta D = \alpha D - 1$ in Step 5). In this case, the new value of D is $D' = D + \alpha D$ and we have $T \leq 5D + \alpha D + 1 + 2\alpha D \leq 5(D + \alpha D) = 5D'$ (since $\alpha D \geq 1$), so item (iii) remains true.

Thus, conditions (i)-(iii) hold in state $\mathbf{A}$, and we always (during all states and transitions between them) have $T \leq 9d_{\text{opt}}$, hence the statement holds. ◀

▶ **Lemma 3.2.** *Assume that we are currently executing* ExpBalancing *on G and that a recursive call is launched after revealing the vertical chord uv with new source u. Let T be the distance traversed before the recursive call. Then, either $T > 9d_{E_*}(s,u)$ or $d_{E_*}(s,v) < d_{E_*}(s,u) + 1$.*

Proof. If $d_{E_*}(s,v) \geq d_{E_*}(s,u) + 1$, following the rules established in Steps 4-6, we will launch a recursive call on the target component of $\{u,v\}$ with new source u. Hence, we will have $d^{\text{Tr}}_{\text{EXP}}(G, E_*) = T + T'$, where $T' \leq 9d_{E_*}(u,t)$ by minimality of G and exp abbreviates ExpBalancing. By way of contradiction, suppose that $T \leq 9d_{E_*}(s,u)$. The optimal offline path P_{opt} necessarily goes through the separator $\{u,v\}$ in graph G and, since $d_{E_*}(s,v) = d_{E_*}(s,u) + 1$, u belongs to some optimal offline path. Consequently, $T + T' \leq 9(d_{E_*}(s,u) + d_{E_*}(u,t)) = 9d_{E_*}(s,t)$. ◀

We are now ready to prove the major contribution of this article.

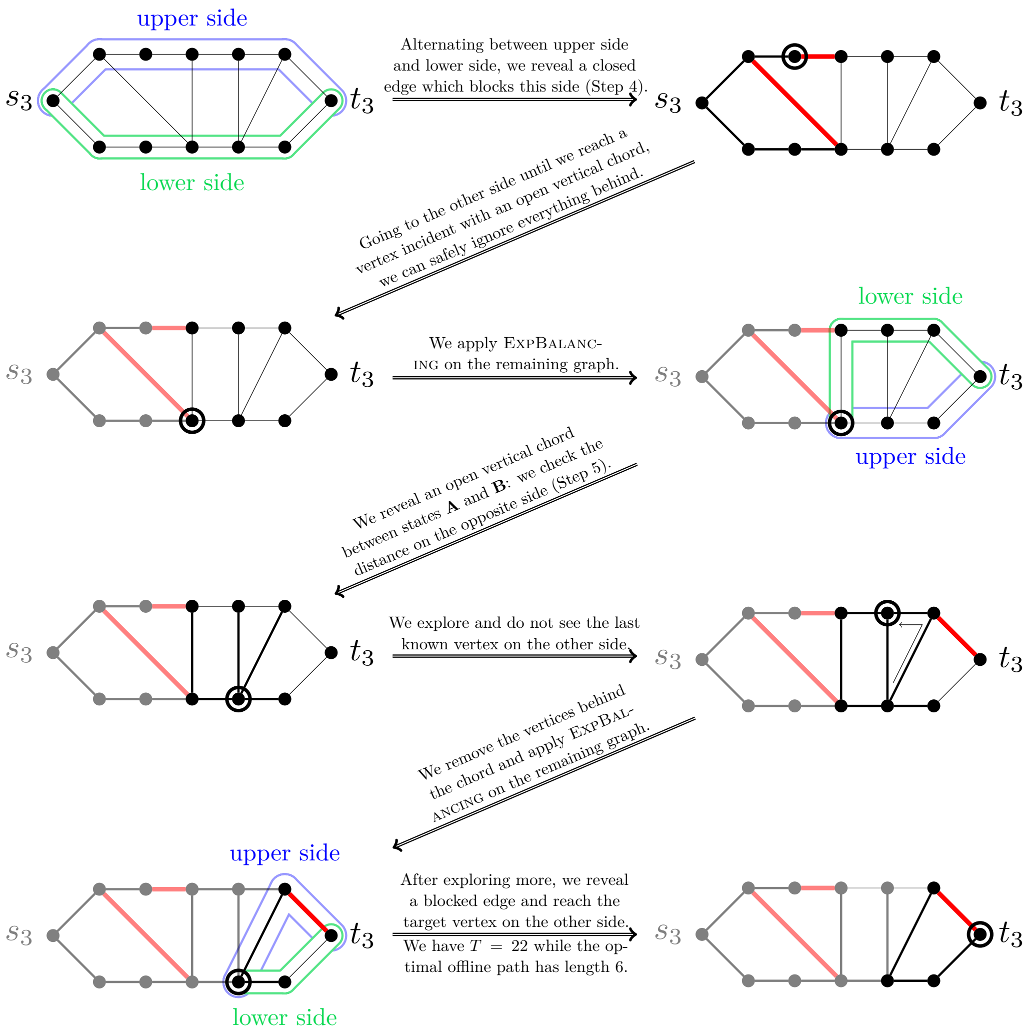

Figure 6 Application of ExpBalancing on the third graph of the decomposition of Figure 2. At each step, the circled vertex is the one we are currently exploring, and we know the status of the bold edges: black is open, red is blocked.

▶ **Theorem 1.1.** *There is a strategy with competitive ratio 9 for unit-weighted outerplanar graphs.*

Proof. A direct consequence of Lemma 3.1 is that, during some attempt, ExpBalancing will launch a recursive call on G (otherwise, it has competitive ratio 9, a contradiction). Let T be the distance traversed before the recursive call. Lemma 3.2 has an important consequence: if we launch a recursive call on the open vertical chord uv with new source u and can guarantee that both $d_{E_*}(s,v) = d_{E_*}(s,u) + 1$ and $T \le 9d_{E_*}(s,u)$, then, we have a contradiction. According to the description of ExpBalancing, a recursive call is launched when we are sure that $d_{E_*}(s,v) = d_{E_*}(s,u) + 1$: this concerns Step 4, Step 5 when $\beta D < \alpha D - 1$ and Step 6.

Assume first that we are blocked on one side between states **A** and **B** in Step 4 (see Figure 4a). We know that $d_{E_*}(s,v) = d_{E_*}(s,u) + 1$ because u is an articulation point of $G \setminus E'_*$. Also, $d_{E_*}(s,u) = D + d_{E_*}(y,u)$. Using Lemma 3.1:

$$\begin{aligned} T &\leq (5D + \alpha D) + (\alpha D + 2D) + d_{E_*}(y,u) \\ &\leq (7 + 2\alpha)D + d_{E_*}(y,u) \\ &\leq 9(D + d_{E_*}(y,u)) \qquad (\alpha \leq 1) \\ &\leq 9d_{E_*}(s,u) \end{aligned}$$

which, by Lemma 3.2, leads to a contradiction.

Assume now that we are blocked on one side between states **C** and **A** in Step 4 (see Figure 4b). Let x' be the last vertex reached at the end of state **A**, we know that $d_{E_*}(s,v) = d_{E_*}(s,u) + 1$ because u is an articulation point of $G \backslash E'_*$ and $d_{E_*}(s,u) = 2D + d_{E_*}(x',u)$. Using Lemma 3.1:

$$\begin{aligned} T &\leq (9D + \alpha D) + (\alpha D + 3D) + d_{E_*}(x',u) \\ &\leq (12 + 2\alpha)D + d_{E_*}(x',u) \\ &\leq 9(2D + d_{E_*}(x',u)) \leq 9d_{E_*}(s,u) \qquad (\alpha \leq 1) \end{aligned}$$

which, by Lemma 3.2, leads to a contradiction.

Assume now that we reveal an open vertical chord uv between states **A** and **B** in Step 5 (see Figure 4c). Recall that $d_{E_*}(s,u) \leq D + \alpha D$, and we explore up to distance $\alpha D - 1$ towards y. There are two possibilities: either we see y by exploring distance βD (with $\beta D < \alpha D - 1$), or we do not see y even if we explore distance $\alpha D - 1$.

If we see y, then, we know that $d_{E_*}(s,u) = d_{E_*}(s,v) + 1$ since going to u through x will yield distance $D + \alpha D$ while going through y and v will yield distance at most $D + \beta D + 2$, and we know that $\beta D < \alpha D - 1$ and $\beta D \geq 0$. So, $d_{E_*}(s,v) = D + \beta D + 1$. Using Lemma 3.1:

$$\begin{aligned} T &\leq (5D + \alpha D) + (1 + 2(\beta D + 1)) \\ &\leq (5 + \alpha + 2\beta)D + 3 \\ &\leq 9(D + \beta D + 1) \leq 9d_{E_*}(s,v) \qquad (\beta < \alpha \leq 1) \end{aligned}$$

which, by Lemma 3.2 leads to a contradiction (the roles of u and v are reversed here, since v is the new source).

If we do not reach y, either by blocked edges or because we have explored distance $\alpha D - 1$ without reaching it, then, we know that $d_{E_*}(s,v) = d_{E_*}(s,u) + 1$. Using Lemma 3.1:

$$\begin{aligned} T &\leq (5D + \alpha D) + (1 + 2(\alpha D - 1) + 1) \\ &\leq (5 + 3\alpha)D \\ &\leq 9(D + \alpha D) \leq 9d_{E_*}(s,u) \qquad (\alpha \leq 1) \end{aligned}$$

which, by Lemma 3.2, leads to a contradiction.

Finally, assume that we reveal an open vertical chord uv between states **C** and **A** after having explored αD vertices in Step 6 (see Figure 4d). Since uv was not revealed before, this implies that the shortest path from s to v goes through u, and so $d_{E_*}(s,v) = d_{E_*}(s,u) + 1$. Using Lemma 3.1:

$$\begin{aligned} T &\leq 9D + \alpha D \\ &\leq 9(D + \alpha D) \leq 9d_{E_*}(s,u) \qquad (\alpha \leq 1) \end{aligned}$$

which, by Lemma 3.2, leads to a contradiction.

All the possible cases lead to contradictions, and so such a G cannot exist. ExpBalancing thus achieves competitive ratio 9 on unit-weighted outerplanar graphs. ◀

3.3 Lower bound 9 for unit-weighted outerplanar graphs

In this subsection we prove that the competitive ratio achieved with the ExpBalancing strategy is optimal on unit-weighted outerplanar graphs.

▶ **Theorem 1.2.** *For any $\varepsilon > 0$, no deterministic strategy achieves competitive ratio $9 - \varepsilon$ on all road maps (G, E_*), where G is a unit-weighted outerplanar graph.*

This result can be obtained by a natural reduction from the linear search problem [9] (or, equivalently, the cow-path problem on two rays [21]). The *linear search problem* is defined as follows: an immobile hider is located on the real line. A searcher starts from the origin and wishes to discover the hider in minimal time. The searcher cannot see the hider until he actually reaches the point at which the hider is located and the time elapsed until this moment is the duration of the game.

This problem reduces to the k-CTP on specific road maps that we call *shell road maps*. The *shell graph* on $2n$ vertices, denoted by Sh_n (see Figure 7), is the graph obtained from a cycle on $2n$ vertices $\{v_0, v_1, \dots, v_{2n-1}\}$ with all possible chords incident with vertex v_n, except v_0v_n. It is clearly outerplanar, and all edge weights are set to 1. In our setting, we shall consider v_0 as the source s and v_n as the target t. We call *shell road maps* the specific road maps (Sh_n, E_*) where E_* is made up only of edges incident with t. Said differently, the traveller cannot be blocked on the outer face on some edge v_iv_{i+1}.

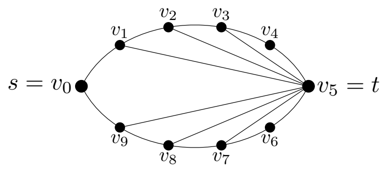

Figure 7 The shell graph on 10 vertices Sh_5.

The shell graph is 2-connected, so it contains an upper side S_1 and a lower side S_2 which can simulate the positive and the negative sides of the real line. The position of the hider will then intuitively correspond to the first encountered open chord to t : if the hider is at position $x > 0$ (resp. $x < 0$), then E_* will contain all $v_it \in E$ except $v_{\lceil x \rceil}t$ (resp. $v_{\lfloor 2n+x \rfloor}t$). In such a way, any strategy for the k-CTP with some competitive ratio r, will give a strategy for the linear search problem with asymptotic competitive ratio $r + \varepsilon$ for any $\varepsilon > 0$. However, it is known that the linear search problem has an optimal ratio of 9 [3] which gives the lower bound we want on the k-CTP. Note however that, in the sketched reduction, small details need to be cared of, for example the distance of the traveller has a unit additive term compared to the searcher on the line (cost of crossing the discovered chord to t). In order to remove any doubt related to these details, we provide in the full version [6] a complete proof of Theorem 1.2 (without reducing to the linear search, but sharing some features with the proof of [3]).

4 The case of arbitrarily weighted outerplanar graphs

Given our results on the unit-weighted case (which give as an easy corollary ratio $9S$ for fixed stretch S), a natural question is whether we can design a deterministic strategy achieving a constant competitive ratio for the more general family of arbitrarily weighted outerplanar graphs. In this section, we prove that this is impossible since, for any constant $C \geq 1$, there exists a weighted outerplanar graph on which the competitive ratio obtained is necessarily greater than C. Let us introduce a sub-family of outerplanar graphs that will be useful here.

▶ **Definition 4.1.** *An outerplanar graph G containing s and t is said to be (s,t)*-unbalanced *if either it is a single st edge or one of its sides contains all vertices V of the graph.*

In other words, an (s,t)-unbalanced outerplanar graph is such that s and t are neighbors on the outer face. While one side (say w.l.o.g. the lower side) contains all vertices, the upper one only contains s and t and simply consists of a single edge st. Thus, such a graph does not have any vertical chord. We show in the remainder that constant competitive ratio cannot be obtained even on weighted (s,t)-balanced outerplanar graphs.

We begin with the definition of a graph transformation $\mathcal{T}$ which takes as input a weighted (s,t)-unbalanced outerplanar graph $H = (V, E, \omega)$, three positive rational values α, C, and η, and an integer N. The construction of the output graph $\mathcal{T}(H, \alpha, C, \eta, N)$ works as follows:

- Create two vertices s and t with an edge st of weight C. This edge will stand as the upper side of the graph.
- Add N copies of the graph αH, where $\alpha H = (V_\alpha, E_\alpha, \omega_\alpha)$ is a graph such that $V_\alpha = V$, $E_\alpha = E$ and $\omega_\alpha(e) = \alpha\omega(e)$ for every edge $e \in E$. These copies are denoted by $\alpha H^{(1)}, \ldots, \alpha H^{(N)}$ and the source/target pair of each $\alpha H^{(j)}$ is denoted by (s_j, t_j).
- Connect in series all copies $\alpha H^{(1)}, \ldots, \alpha H^{(N)}$ from s to t in order to form the lower side of the graph, using their source/target as input/output vertices. In brief, merge s with s_1, t_i with s_{i+1} for $i \in \{1, \ldots, N-1\}$, and t_N with t.
- Add all edges $t_j t$ for $1 \leq j \leq N-1$ with weight η.

Figure 8 illustrates the graph $\mathcal{T}(H, \alpha, C, \eta, N)$ obtained. Observe that it is an (s,t)-unbalanced outerplanar graph because the lower side of each αH contains all its own vertices. Therefore, all vertices of $\mathcal{T}(H, \alpha, C, \eta, N)$ lie on its lower side. We also set $t_0 = s$.

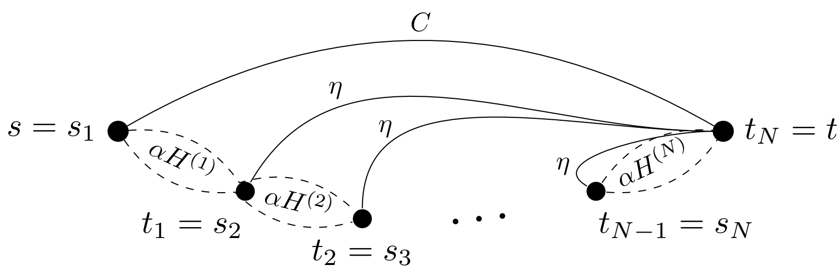

Figure 8 The graph $\mathcal{T}(H, \alpha, C, \eta, N)$ with its outerplanar embedding.

For the remainder, we define a trivial arithmetic sequence generating all positive half-integers: for any integer $i \geq 0$, let $C_i = \frac{1}{2} + i$. For any value C_i, we are able to construct a collection of road maps for which ratio C_i cannot be achieved by any deterministic strategy.

▶ **Proposition 4.2 (*).** *For any nonnegative integer i, there exists a family $\mathcal{R}_i$ of road maps which satisfies the following properties:*

- *all the road maps of $\mathcal{R}_i$ are defined on the same weighted (s,t)-unbalanced outerplanar graph,*
- *no deterministic strategy can achieve ratio C_i on family $\mathcal{R}_i$.*

Sketch of the proof. By induction, we assume that the property holds for some $i \geq 1$. We focus on some graph $H_{i+1} = \mathcal{T}(H_i, \alpha, C_i, \eta, N)$. The reasoning consists in a trade-off on the optimal distance between s and the last visited t_j vertex on the lower side, denoted by t_q. If $d_{E_*}(s, t_q)$ is at least C_i, then it appears that the st edge is the optimal offline path and we realize that exploring the lower side was too costly. We deduce that the obtained competitive ratio is necessarily greater than $C_i + 1$ using the induction hypothesis. Otherwise, if $d_{E_*}(s, t_q) < C_i$, then it means that we did not explore far enough on the lower side, hence the optimal offline path passes through it. We obtain the same conclusion that the ratio is greater than $C_i + 1$. See full version [6] for the proof.

Hence, no deterministic strategy can achieve constant competitive ratio on weighted outerplanar graphs since integer i can take arbitrarily large values:

▶ **Theorem 1.3.** *There is no constant C, independent from G and k, such that a deterministic strategy achieves competitive ratio C on all road maps (G, E_*) where G is a weighted outerplanar graph.*

Proof. By contradiction, for any $C \geq 1$, apply Proposition 4.2 on $i = \lceil C \rceil$. ◀

5 Perspectives

We highlighted a non-trivial unit-weighted family of graphs (outerplanar) for which there exists a deterministic strategy with constant competitive ratio 9, which is optimal. However, we proved that no constant competitive ratio can be achieved for arbitrarily weighted outerplanar graphs. Several questions arise.

Since some sub-families of outerplanar graphs have constant competitive ratio in the weighted case (trees and cycles, which imply cacti from Lemma 2.2) while a very close super-family admits the general bound $2k + 1$ in the unit-weighted case (planar of treewidth 2), a natural question is to investigate where the competitive gaps lie in both cases. For the unit-weighted case, future research could focus on the natural extension of p-outerplanar graphs [4], with p successive outer faces, in order to generalize constant competitiveness.

To achieve constant competitive ratio on arbitrarily weighted graphs, a good candidate could be graphs with bounded-sized minimal edge (s, t)-cuts, for which ratio $\sqrt{2}k + O(1)$ is known [12]. Observe that our construction $\mathcal{T}$ which disproves constant ratio increases the size of edge (s, t)-cuts. We conjecture that there exists a polynomial-time deterministic strategy achieving constant competitive ratio on graphs with edge (s, t)-cuts of bounded size.

References

1 V. Aksakalli, O. F. Sahin, and I. Ari. An AO* based exact algorithm for the canadian traveler problem. *INFORMS Journal on Computing*, 28(1):96–111, 2016. `doi:10.1287/IJOC.2015.0668`.

2 A. F. Alkaya, S. Yildirim, and V. Aksakalli. Heuristics for the Canadian Traveler Problem with neutralizations. *Comput. Ind. Eng.*, 159:107488, 2021. `doi:10.1016/J.CIE.2021.107488`.

3 R. A. Baeza-Yates, J. C. Culberson, and G. J. E. Rawlins. Searching in the plane. *Inf. Comput.*, 106(2):234–252, 1993. `doi:10.1006/INCO.1993.1054`.

4 B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.

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

6 L. Beaudou, P. Bergé, V. Chernyshev, A. Dailly, Y. Gerard, A. Lagoutte, V. Limouzy, and L. Pastor. The Canadian Traveller Problem on outerplanar graphs. *CoRR*, abs/2403.01872, 2024. `doi:10.48550/arXiv.2403.01872`.

7 A. Beck and D. J. Newman. Yet more on the linear search problem. *Isr. J. Math.*, 8(4):419–429, 1970.

8 J. Becker and R. Batta. Canadian prize collection problem. *Military Operations Research*, 28(2):55–92, 2023.

9 R. Bellman. Problem 63-9, an optimal search. *SIAM review*, 5(3):274–274, 1963.

10 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. doi:10.1007/S10878-013-9634-8.

11 P. Bergé, J. Desmarchelier, W. Guo, A. Lefebvre, A. Rimmel, and J. Tomasik. Multiple canadians on the road: minimizing the distance competitive ratio. *J. Comb. Optim.*, 38(4):1086–1100, 2019. doi:10.1007/S10878-019-00438-6.

12 P. Bergé and L. Salaün. The influence of maximum (s, t)-cuts on the competitiveness of deterministic strategies for the Canadian Traveller Problem. *Theor. Comput. Sci.*, 941:221–240, 2023. doi:10.1016/J.TCS.2022.11.017.

13 Z. Bnaya, A. Felner, and S. E. Shimony. Canadian traveler problem with remote sensing. In *Proc. of IJCAI*, pages 437–442, 2009.

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

15 H. Chan, J. Chang, H. Wu, and T. Wu. The k-Canadian Traveller Problem on Equal-Weight Graphs. *Proc. of WCMCT*, pages 135–137, 2015.

16 E. D. Demaine, Y. Huang, C. Liao, and K. Sadakane. Canadians Should Travel Randomly. *Proc. of ICALP*, pages 380–391, 2014. doi:10.1007/978-3-662-43948-7_32.

17 R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

18 P. Eyerich, T. Keller, and M. Helmert. High-quality policies for the Canadian Traveler's Problem. In *Proces. of AAAI*, pages 51–58. AAAI Press, 2010. doi:10.1609/AAAI.V24I1.7542.

19 D. Fried, S. E. Shimony, A. Benbassat, and C. Wenner. Complexity of canadian traveler problem variants. *Theor. Comput. Sci.*, 487:1–16, 2013. doi:10.1016/J.TCS.2013.03.016.

20 N. Hahn and M. Xefteris. The covering canadian traveller problem revisited. In *Procs. of MFCS*, volume 272 of *LIPIcs*, pages 53:1–53:12, 2023. doi:10.4230/LIPICS.MFCS.2023.53.

21 M.-Y. Kao, J. H. Reif, and S. T. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Information and computation*, 131(1):63–79, 1996. doi:10.1006/INCO.1996.0092.

22 C. Liao and Y. Huang. The covering canadian traveller problem. *Theor. Comput. Sci.*, 530:80–88, 2014. doi:10.1016/J.TCS.2014.02.026.

23 L. V. Lita, J. Schulte, and S. Thrun. A system for multi-agent coordination in uncertain environments. In *Procs. of AGENTS*, pages 21–22. ACM, 2001. doi:10.1145/375735.375806.

24 C. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theor. Comput. Sci.*, 84(1):127–150, 1991. doi:10.1016/0304-3975(91)90263-2.

25 D. Shiri and F. S. Salman. On the online multi-agent $O-D$ k-Canadian Traveler Problem. *J. Comb. Optim.*, 34(2):453–461, 2017. doi:10.1007/S10878-016-0079-8.

26 S. Westphal. A note on the k-Canadian Traveller Problem. *Inform. Proces. Lett.*, 106(3):87–89, 2008. doi:10.1016/J.IPL.2007.10.004.

27 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. doi:10.1007/S10878-008-9156-Y.