Exploration of Graphs with Excluded Minors

Júlia Baligács
Technische Universität Darmstadt, Germany

Yann Disser
Technische Universität Darmstadt, Germany

Irene Heinrich
Technische Universität Darmstadt, Germany

Pascal Schweitzer
Technische Universität Darmstadt, Germany

Abstract

We study the online graph exploration problem proposed by Kalyanasundaram and Pruhs (1994) and prove a constant competitive ratio on minor-free graphs. This result encompasses and significantly extends the graph classes that were previously known to admit a constant competitive ratio. The main ingredient of our proof is that we find a connection between the performance of the particular exploration algorithm BLOCKING and the existence of light spanners. Conversely, we exploit this connection to construct light spanners of bounded genus graphs. In particular, we achieve a lightness that improves on the best known upper bound for genus $g \geq 1$ and recovers the known tight bound for the planar case ($g = 0$).

2012 ACM Subject Classification Theory of computation → Online algorithms; Theory of computation → Sparsification and spanners; Mathematics of computing → Graphs and surfaces

Keywords and phrases online algorithms, competitive analysis, graph exploration, graph spanners, minor-free graphs, bounded genus graphs

Digital Object Identifier 10.4230/LIPIcs.ESA.2023.11

Related Version *Full Version*: `http://arxiv.org/abs/2308.06823`

1 Introduction

We study a classic online graph exploration problem that was first proposed by Kalyanasundaram and Pruhs in 1994 [29]. In this setting, a single agent needs to systematically traverse an initially unknown, undirected, connected graph with non-negative edge weights. Upon visiting a new vertex, the agent learns the unique identifiers of all adjacent vertices and the weights of the corresponding edges. The cost incurred when traversing an edge is simply its weight. The objective in online graph exploration is to visit all vertices of the graph and return to the starting vertex while minimizing the total cost.

The performance of a (deterministic) online algorithm $\textsc{Alg}$ is measured in terms of competitive analysis. That is, given a graph G and starting vertex v of G, we compare the cost $\textsc{Alg}(G, v)$ of the traversal it produces to the cost of an offline optimum traversal $\textsc{Opt}(G)$. Note that the optimum cost corresponds to the length of a shortest TSP tour of G and does not depend on v. We say that $\textsc{Alg}$ is *(strictly) ρ-competitive* for a class of graphs if $\textsc{Alg}(G, v) \leq \rho \cdot \textsc{Opt}(G)$ for every graph G in the class and every vertex v of G. The *(strict) competitive ratio* of an algorithm $\textsc{Alg}$ is given by $\inf\{\rho : \textsc{Alg} \text{ is } \rho\text{-competitive}\}$.

Kalyanasundaram and Pruhs [29] posed the following question: *Is there a deterministic algorithm for online graph exploration with a constant competitive ratio?* Several algorithms were proposed with a competitive ratio of $\mathcal{O}(\log(n))$ [31, 36], where n is the number of vertices, but better competitive ratios are only known for restricted classes of graphs [29, 31, 33]. The best known lower bound on the competitive ratio is 10/3 [5]. In particular, the original question of Kalyanasundaram and Pruhs remains open.

31st Annual European Symposium on Algorithms (ESA 2023).
Editors: Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman; Article No. 11; pp. 11:1–11:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We formalize a connection between the performance of the particular exploration algorithm Blocking and the existence of light spanners. Spanners were introduced in 1989 by Peleg and Schäffer [35] and have been instrumental in the development of approximation algorithms, particularly for TSP [3, 8, 9]. Here, a subgraph $H = (V, E_H)$ of a connected, undirected graph $G = (V, E)$ with edge weights $w\colon E \to \mathbb{R}_{\geq 0}$ is called a $(1+\varepsilon)$-*spanner* of G if $d_H(u, v) \leq (1+\varepsilon)\, d_G(u, v)$ for all $u, v \in V$, where d_H and d_G denote the shortest-path distance in H and G, respectively. Then, H has *stretch* at most $(1+\varepsilon)$ and its *lightness* is $w(H)/w(\mathrm{MST})$, where $w(H) := \sum_{e \in E_H} w(e)$ and MST denotes a minimum spanning tree of G.

We show that the online graph exploration algorithm Blocking has a constant competitive ratio on every class of graphs that admits spanners of constant lightness for a fixed stretch. Prominent graph classes with this property are the classes with a forbidden minor [9]. We thus, in particular, obtain a constant competitive ratio for online graph exploration on all graph classes excluding a minor. They encompass many other important classes, such as graphs of bounded genus or bounded treewidth. Overall, this result subsumes and significantly extends all previously known graph classes for which a competitive ratio of $o(\log(n))$ was known.

Regarding research for graph spanners, results typically revolve around the existence of good, in particular light, spanners. For example, the Erdős girth conjecture [19] is equivalent to a lower bound of $\Omega(n^{1/k})$ on the best lightness of a $(2k-1)$-spanner in unweighted graphs. While this conjecture remains unresolved, a nearly matching upper bound was proven by Chechik and Wulff-Nilsen [11]. Various constant upper bounds on the lightness are known for restricted classes of graphs [2, 9, 12, 24]. Our newly discovered connection to graph exploration also allows us to contribute an improved upper bound for graphs of bounded genus using the ideas given in [31].

Our results. We significantly expand the class of graphs on which the exploration problem admits a constant-competitive algorithm.

▶ **Theorem 1.** *For every graph H and constant $\delta > 0$, there is a constant c (depending on H and δ) such that* Blocking$_\delta$ *is c-competitive on H-minor-free graphs.*

The technical contribution of this work is a new-found connection between graph spanners and the performance of the exploration algorithm Blocking$_\delta$ (see Section 2.1) introduced by Megow et al. [31] based on an algorithm of Kalyanasundaram and Pruhs [29]. This connection will allow us to prove Theorem 1.

Prior to our work, the largest class of graphs which was known to admit a constant-competitive algorithm was the class of bounded genus graphs [31]. As an aside, we obtain a slightly stronger bound also for bounded genus graphs (cf. Corollary 13).

So far, Blocking$_\delta$ was only studied for constant choices of the parameter δ, i.e., independent of the number of vertices n. It is known that its competitive ratio is at least $\Omega(n^{1/(4+\delta)})$ if δ is a constant [31]. This naturally raises the question of whether improvement is possible if δ may depend on n. We obtain the following results.

▶ **Theorem 2.** *Blocking*$_{\log(n)}$ *is $O(\log(n))$-competitive.*

This shows that Blocking$_{\log(n)}$ achieves the best previously known competitiveness. We complement this with the following lower bounds.

▶ **Theorem 3.** *The competitive ratio of* Blocking$_\delta$*, where $\delta = \delta(n) > 0$, is at least*

a) $\Omega(\log(n)/\log(\log(n)))$,

b) $\Omega(\log(n))$ *for $\delta \in o(\log(n)/\log\log(n))$ as well as for $\delta \in \Omega(\log(n))$.*

In particular, this shows that there is no δ such that BLOCKING$_\delta$ is constant-competitive, but it remains open, whether there is a choice of δ for which the algorithm is $o(\log(n))$-competitive.

Next, we exploit the connection between spanners and exploration in reverse to derive the existence of good spanners in bounded genus graphs.

▶ **Theorem 4.** *For every $\varepsilon > 0$, the greedy $(1+\varepsilon)$-spanner of a graph of genus g has lightness at most $\left(1+\frac{2}{\varepsilon}\right)\left(1+\frac{2g}{1+\varepsilon}\right)$.*

Prior to our work, the best known bound was due to Grigni [24] who showed that every graph of genus $g \geq 1$ contains a $(1+\varepsilon)$-spanner of lightness $1+\frac{12g-4}{\varepsilon}$. Moreover, it is already known that planar graphs, i.e., graphs of genus 0, contain $(1+\varepsilon)$-spanners of lightness $1+\frac{2}{\varepsilon}$ and that this is best possible [2]. This means that Theorem 4 gives a tight bound in the case $g = 0$ and extrapolates this bound to graphs of larger genus.

Related work. Kalyanasundaram and Pruhs [29] introduced the online graph exploration problem and gave a constant-competitive algorithm for planar graphs. Megow, Mehlhorn and Schweitzer [31] revisited the algorithm, addressed some technical intricacies, and proposed their reinterpretation BLOCKING$_\delta$, which we also consider in this paper. They expanded the result by Kalyanasundaram and Pruhs and showed that the algorithm is constant-competitive on bounded genus graphs. Moreover, they suggested a new algorithm hDFS and showed that it is constant-competitive on graphs with a bounded number of different weights and $O(\log(n))$-competitive on general graphs.

Another very natural approach for exploration is the *Nearest Neighbor* algorithm, which, in each step, explores the unvisited vertex nearest to the current location. This algorithm has been studied extensively as a TSP heuristic. Rosenkrantz, Stearns and Lewis were able to show that its competitive ratio is $\Theta(\log(n))$ [36]. It turned out that the lower bound of $\Omega(\log(n))$ is already achieved on unweighted planar graphs [28] and on trees [23]. Eberle et al. [18] revisited the algorithm with learning augmentation.

In addition to planar and bounded genus graphs, the exploration problem has been studied on many more graph classes. For example, Miyazaki, Morimoto and Okabe were able to show that the competitive ratio of the exploration problem is $(1+\sqrt{3})/2$ on cycles and 2 on unweighted graphs. Other examples of such graph classes are tadpole graphs [10], unicyclic graphs [23], and cactus graphs [23].

Currently, the best known lower bound for the graph exploration problem is 10/3 which was shown by Birx, Disser, Hopp, and Karousatou [5]. Their construction builds on a previously known lower bound of 2.5 shown by Dobrev, Královič, and Markou [17]. Since the construction by Birx et al. is planar, the lower bound of 10/3 even holds when the problem is restricted to planar graphs.

Several other settings of the exploration problem have been studied, such as exploration on directed graphs [1, 13, 22, 21] or exploration with a team of agents [14, 15, 16]. Another problem which is closely related to graph exploration is online TSP, where a single agent has to serve requests appearing over time in a known graph [6, 7].

Through the connection with spanners, we are concerned with the existence of light spanners for a given stretch. Examples of graph classes where the worst-case lightness does not depend on the number of vertices include planar graphs [2], bounded genus graphs [24], apex graphs [26], bounded pathwidth graphs [25], bounded treewidth graphs [12], and minor-free graphs [9]. Our results rely on the existence of light spanners for minor-free graphs [9] and improve on the lightness for bounded genus graphs. In particular, we study the lightness of the so-called greedy spanner [2] for graphs of bounded genus. It was shown by Filtser and

Solomon [20] that this spanner construction is existentially optimal for every class of graphs closed under taking subgraphs, which means that the optimal lightness guarantee on any such class is achieved by the greedy spanner.

Light and sparse spanners have applications in various fields. Most importantly, spanners were used to give polynomial-time approximation schemes (PTAS) for the travelling salesperson problem for various graph classes [3, 8, 9]. Note that the difference between approximations for TSP and online exploration is that, in our setting, the tour is computed on-the-fly. Indeed, in comparison to our online setting, we desire a constant approximation for an arbitrary constant, which in the TSP setting is easily obtained by traversing a minimum spanning tree twice. On the other hand, in the online setting, we are not concerned with efficiency of the algorithms which is crucial in the TSP setting. Other fields of application of spanners include distributed systems [4], routing [38], or computational biology [37].

2 The online graph exploration problem on minor-free graphs

In this section, we prove new upper bounds for $\textsc{Blocking}_\delta$ on H-minor-free graphs (Theorem 1) and for general graphs (Theorem 2). To this end, we begin by introducing the algorithm $\textsc{Blocking}_\delta$ proposed by Megow et al. [31] based on the work of Kalyanasundaram and Pruhs [29].

2.1 The algorithm Blocking

During the execution of an online graph exploration algorithm, a vertex is *explored* if it has been visited by the agent. A neighbor of an explored vertex is a *learned* vertex. An edge is a *boundary edge* if exactly one of its endpoints is explored. By convention, we denote boundary edges $e = (u, v)$ such that u is explored and v is unexplored. A path is *internally explored* if each of its internal vertices is explored. Given two learned vertices x and y, we set the distance $d(x, y)$ to be the length of a shortest internally explored path linking x with y. In particular, the distance may decrease during execution.

▶ **Definition 5** (Kalyanasundaram and Pruhs [29]). *Given some $\delta > 0$, we say that a boundary edge $e = (u, v)$ is* δ-blocked *if there is another boundary edge $e' = (u', v')$ such that $w(e') < w(e)$ and $d(u, v') \leq (1 + \delta)w(e)$.*

The rough idea of Blocking is to perform a depth-first-traversal while ignoring all blocked edges. Whenever a previously blocked edge turns unblocked, the agent moves to and explores one such edge, and initiates a DFS-traversal from its new position. Blocking is formally specified in Algorithm 1. It is executed on an undirected, weighted, connected, and initially unexplored graph $G = (V, E, w)$ and takes as input a vertex v of G, denoting the current position of the agent. The algorithm follows a recursive DFS-like structure and the input of the initial invocation is the start vertex.

Algorithm 1 $\textsc{Blocking}_\delta(v)$ [29, 31].

```
while there is a boundary edge e = (y, x) that is not δ-blocked and such that y = v
  or e was previously blocked by some edge (u, v) do
    traverse a shortest internally explored path from v to y
    traverse e
    Blocking_δ(x)
    traverse a shortest internally explored path from x to v
```

Observe that the algorithm is correct, i.e., every vertex is explored: Assume, for the sake of contradiction, that some vertex remains unexplored when the algorithm terminates, i.e., there are still boundary edges. Let $e = (u, v)$ be a boundary edge of minimum weight. Then, e is not δ-blocked. Therefore, either the exploration of u should have triggered the exploration of v, or v should have been explored at the last point in time the edge turned unblocked.

2.2 Key properties of Blocking

Throughout the remainder of Section 2, let $G = (V, E, w)$ be a graph, $n = |V|$ be its number of vertices, v the given start vertex of G, and $\delta = \delta(n) > 0$. We analyze the performance of Blocking_δ on G, i.e., we estimate its total cost $W_{\text{Blocking}}(G, v, \delta)$. For this, let B be the set of boundary edges taken by Blocking_δ, i.e., the edges traversed during the execution of line 3.

Note that the total cost of the offline optimum is bounded from below by the weight of a minimum spanning tree $w(\text{MST})$ and from above by $2w(\text{MST})$. That is, to show that Blocking_δ is ρ-competitive, it suffices to show $W_{\text{Blocking}}(G, v, \delta) \leq \rho \cdot w(\text{MST})$.

▶ **Observation 6** (Megow et al. [31]). *We have $W_{\text{Blocking}}(G, v, \delta) \leq 2(\delta + 2)w(B)$.*

Proof. We charge all cost incurred in lines 2,3, and 5 to the corresponding boundary edge $e \in B$. Note that the cost in line 2 is at most $(1 + \delta)w(e)$, because either we have $y = v$ such that $d_G(v, y) = 0$, or e was blocked by an edge (u, v), which implies $d_G(y, v) \leq (1+\delta)w(e)$. The cost in line 3 is $w(e)$ and the cost in line 5 is at most the sum of the cost in lines 2 and 3. Therefore, each edge e in B is charged at most $2(\delta + 2)w(e)$. ◀

In our subsequent analysis, we will frequently use a minimum spanning tree with a particular property. For this, in what follows, let MST_B be a minimum spanning tree of G that maximizes the number of edges in $\text{MST}_B \cap B$. As pointed out in [31], cycles in $B \cup \text{MST}_B$ are long relative to the weight of the edges they contain. Specifically, the following holds.[1]

▶ **Lemma 7.** *Let C be a cycle in $B \cup \text{MST}_B$ and e be an edge of C. Then,*

$$w(C \setminus \{e\}) > (1 + \delta)w(e).$$

Proof. It suffices to show the assertion for an edge of maximum weight in C. We first show that this edge must be in B, i.e., $\text{argmax}\{w(e) : e \in C\} \subseteq B$:

Assume otherwise and let $e = (u, v) \in \text{argmax}\{w(e) : e \in C\} \cap (\text{MST}_B \setminus B)$. Removing e from MST_B separates MST_B into two connected components. In particular, u and v are in different components. Start walking in $C \setminus \{e\}$ from u to v and let e' be the first edge that leads from u's connected component in $\text{MST}_B \setminus \{e\}$ to v's connected component. Then, $e' \in B \setminus \text{MST}_B$ and by maximality of e, we have $w(e') \leq w(e)$. Therefore, replacing e by e' in MST_B gives another spanning tree of weight at most $w(\text{MST}_B)$. This new spanning tree has one more edge in common with B than MST_B. This contradicts the choice of MST_B, so that we can assume from now on $\text{argmax}\{w(e) : e \in C\} \subseteq B$, i.e., every edge in $\text{argmax}\{w(e) : e \in C\}$ is *charged*, i.e., the edge is traversed in some exectution of line 3 of the algorithm.

[1] The assertion of Lemma 7 implies Claim 1 in [31], which only concerns edges not in the minimum spanning tree. However, there is a subtle flaw in the proof of Claim 1 in [31]. In fact, in that proof, it is not clear that when we replace the edge e' with an edge of the fixed MST, we again obtain a minimum spanning tree. In any case, the argument above rectifies this.

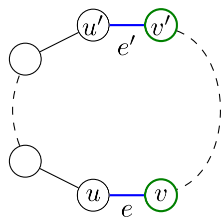

Figure 1 Illustration of Lemma 7: The black vertices are explored and the green vertices (v and v') are unexplored. The blue edges (e and e') are boundary edges.

Let $e = (u, v)$ be the edge in $\operatorname{argmax}\{w(e) : e \in C\}$ that is charged last. At the time e is traversed, it is a boundary edge, so that u is explored but v is not yet explored. Start walking in $C \setminus \{e\}$ from u to v and let $e' = (u', v')$ be the first edge leading from an explored vertex u' to an unexplored vertex v', i.e., e' is another boundary edge in C (cf. Figure 1).

Next, we show that $w(e') < w(e)$: Assume otherwise. By maximality of e, this means $w(e') = w(e)$ so that $e' \in \operatorname{argmax}\{w(e) : e \in C\}$. But then, we also have $e' \in B$. This contradicts the fact that e is the edge in $\operatorname{argmax}\{w(e) : e \in C\}$ that is charged last.

Summing up, we have shown the following facts: Upon exploration of $e = (u, v)$, there is another boundary edge $e' = (u', v')$ in C with $w(e') < w(e)$. Since e is not blocked, this implies

$$w(C \setminus \{e\}) \geq d(u, v') > (1 + \delta)w(e). \quad ◀$$

2.3 Connection to spanners

Next, we investigate how the performance of $\textsc{Blocking}_\delta$ is related to graph spanners. For this, note that Lemma 7 can be reformulated as follows.

▶ **Lemma 8.** *No proper subgraph of $B \cup \mathrm{MST}_B$ is a $(1 + \delta)$-spanner of $B \cup \mathrm{MST}_B$.*

The lemma relates spanners to the behavior of $\textsc{Blocking}_\delta$. However, we need to take note that the lemma applies to $B \cup \mathrm{MST}_B$ rather than the original graph G. A *monotone graph class* is a class of graphs $\mathcal{G}$ closed under taking subgraphs, i.e., if $G \in \mathcal{G}$ and H is a subgraph of G, then also $H \in \mathcal{G}$. Given a graph G, we define $\textsc{OptSpan}_\delta(G)$ as the minimum lightness of a $(1 + \delta)$-spanner of G. Moreover, we set $\textsc{OptSpan}_\delta(\mathcal{G}) := \sup\{\textsc{OptSpan}_\delta(G) : G \in \mathcal{G}\}$ to be the supremum over all graphs in $\mathcal{G}$.

▶ **Theorem 9.** *For every monotone graph class $\mathcal{G}$ and every $\delta = \delta(n) > 0$, the algorithm $\textsc{Blocking}_\delta$ is $(2(\delta + 2) \cdot \textsc{OptSpan}_\delta(\mathcal{G}))$-competitive on $\mathcal{G}$.*

Proof. Let $G \in \mathcal{G}$. We have

$$W_{\textsc{Blocking}}(G, v, \delta) \overset{\text{Obs } 6}{\leq} 2(\delta + 2)w(B) \leq 2(\delta + 2)w(B \cup \mathrm{MST}_B). \tag{1}$$

Since $B \cup \mathrm{MST}_B$ is a subgraph of G, we have $B \cup \mathrm{MST}_B \in \mathcal{G}$. By Lemma 8, the only $(1 + \delta)$-spanner of $B \cup \mathrm{MST}_B$ is $B \cup \mathrm{MST}_B$ itself. Therefore,

$$w(B \cup \mathrm{MST}_B) \leq \textsc{OptSpan}_\delta(B \cup \mathrm{MST}_B) \cdot w(\mathrm{MST}_B) \leq \textsc{OptSpan}_\delta(\mathcal{G}) \cdot w(\mathrm{MST}_B). \tag{2}$$

Combined, we obtain

$$W_{\textsc{Blocking}}(G, v, \delta) \overset{(1)}{\leq} 2(\delta + 2)w(B \cup \mathrm{MST}_B) \overset{(2)}{\leq} 2(\delta + 2) \cdot \textsc{OptSpan}_\delta(\mathcal{G}) \cdot w(\mathrm{MST}_B). \quad ◀$$

The theorem puts us in a position to leverage results on the lightness of spanners in order to draw conclusions regarding the competitive ratio of $\textsc{Blocking}_\delta$. For example, it has been shown that every planar graph contains a $(1+\delta)$-spanner of lightness at most $1+\frac{2}{\delta}$ [2]. Feeding this into Theorem 9, we conclude that $\textsc{Blocking}_\delta$ is $2(\delta+2)(1+2/\delta)$-competitive on planar graphs. This agrees with the bound proven in [29]. However, more generally, bounded genus graphs have light spanners. In fact, in Section 3.3, we show that every graph of genus at most g contains a $(1+\delta)$-spanner of lightness at most $\left(1+\frac{2}{\delta}\right)\left(1+\frac{2g}{1+\delta}\right)$ (Theorem 4). From this, we obtain the following.

▶ **Corollary 10.** *$\textsc{Blocking}_\delta$ is $2(\delta+2)\left(1+\frac{2}{\delta}\right)\left(1+\frac{2g}{1+\delta}\right)$-competitive on graphs of genus at most g.*

Even more generally, it is known that H-minor-free graphs have light spanners [9]. Specifically, every H-minor-free graph contains a $(1+\delta)$-spanner of lightness $O\left(\frac{\sigma_H}{\delta^3}\log\left(\frac{1}{\delta}\right)\right)$ where $\sigma_H = |V(H)|\sqrt{\log|V(H)|}$. This yields a constant competitive ratio for $\textsc{Blocking}_\delta$ on H-minor-free graphs as follows.

▶ **Corollary 11.** *$\textsc{Blocking}_\delta$ is $2(\delta+2)\cdot O\left(\frac{\sigma_H}{\delta^3}\log\left(\frac{1}{\delta}\right)\right)$-competitive on H-minor-free graphs where $\sigma_H = |V(H)|\sqrt{\log|V(H)|}$.*

There are also strong bounds for general graphs. Given a graph G with n vertices and an integer $k \geq 1$ and $\varepsilon \in (0,1)$, G contains a $(2k-1)(1+\varepsilon)$-spanner of lightness $O_\varepsilon\left(n^{1/k}\right)$ [11], where the notation O_ε indicates that the constant factor hidden in the O-notation depends on ε. This gives us the following.

▶ **Corollary 12.** *Given an integer $k = k(n) \geq 1$ and $\varepsilon \in (0,1)$, $\textsc{Blocking}_{(2k-1)(1+\varepsilon)}$ is $2((2k-1)(1+\varepsilon)+2)\cdot O_\varepsilon\left(n^{1/k}\right)$-competitive on every graph.*

In particular, by suitably choosing δ, we obtain the following.[2]

▶ **Corollary 13.**

a) *$\textsc{Blocking}_2$ is $16(1+\frac{2}{3}g)$-competitive on graphs of genus at most g.*

b) *For every constant $\delta > 0$ and every graph H, $\textsc{Blocking}_\delta$ is constant-competitive on H-minor-free graphs.*

c) *$\textsc{Blocking}_{\log(n)}$ is $O(\log(n))$-competitive on every graph.*

For the case of planar graphs, part a) matches the best known bounds on planar graphs [29, 31]. For general surfaces, it slightly improves on the best known bound of $16(1+2g)$ on bounded genus graphs [31]. Part b) is the first constant bound on minor-free graphs, and part c) is the first $O(\log(n))$ bound for Blocking.

2.4 Lower bounds for Blocking

Next, we investigate lower bounds for Blocking when δ is allowed to depend on the input size. In [31], it was shown that the competitive ratio of $\textsc{Blocking}_\delta$ on general graphs is at least $\Omega(n^{1/(\delta+4)})$ when δ is a constant. We begin by observing that this can be generalized to non-constant δ that are not too large.

▶ **Observation 14.** *Suppose $\delta = \delta(n) > 0$ such that $\delta^{2\delta+8} = o(n)$. Then, the competitive ratio of $\textsc{Blocking}_\delta$ is at least $\Omega(\delta \cdot n^{1/(\delta+4)})$.*

[2] All missing proofs are deferred to the full version.

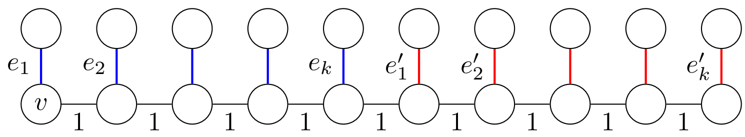

Figure 2 Illustration of the lower bound construction for Blocking$_\delta$ (Lemma 15). The light edges (depicted in blue) are of weight 1 and the heavy edges (depicted in red) are of weight $\frac{k+1}{\delta+1}$.

Note that $2\delta + 8 \le \log(n)/\log(\log(n))$ implies that

$$\delta^{2\delta+8} \le \left(\frac{\log(n)}{\log(\log(n))}\right)^{\frac{\log(n)}{\log(\log(n))}} = \left(\frac{1}{\log(\log(n))}\right)^{\frac{\log(n)}{\log(\log(n))}} \cdot e^{\log(\log(n))\frac{\log(n)}{\log(\log(n))}}$$
$$= \left(\frac{1}{\log(\log(n))}\right)^{\frac{\log(n)}{\log(\log(n))}} \cdot n = o(n),$$

i.e., the prerequisites of Observation 14 are fulfilled. Moreover, $\Omega(\delta n^{1/(\delta+4)}) \ge \Omega(\log(n))$ for every $\delta = \delta(n)$. Therefore, Observation 14 shows that Blocking$_\delta$ has competitive ratio in $\Omega(\log(n))$ whenever $\delta = o(\log(n)/\log\log(n))$. In particular, this shows the first part of Theorem 3b.

Next, we give another lower bound which shows that the parameter δ cannot be chosen too large either (the second part of Theorem 3b).

▶ **Lemma 15.** *Suppose $\delta = \delta(n) \in (0, \frac{n-4}{4})$. The competitive ratio of Blocking$_\delta$ is at least $\Omega(\delta)$, even on trees.*

Proof sketch. It is not difficult to check that, on the graph illustrated in Figure 2, the cost of Blocking is asymptotically δ times the cost of the offline optimum. A complete proof can be found in the full version. ◀

To conclude our lower bound arguments for Blocking$_\delta$, observe that, for $\delta \ge \frac{n-4}{4}$, the behavior of Blocking$_\delta$ closely resembles the behavior of the algorithm hDFS [31]. In fact, it is not difficult to check that, on the lower bound construction for hDFS given in [31, Theorem 5], after proceeding to edges of weight more than 16, Blocking$_\delta$ takes the exact same route as hDFS, if $\delta \ge \frac{n-4}{4}$. Therefore, we obtain the following.

▶ **Observation 16.** *For $\delta \ge \frac{n-4}{4}$, the competitive ratio of Blocking$_\delta$ is at least $\Omega(\log(n))$.*

We can now combine the lower bound constructions from this section to prove Theorem 3.

Proof of Theorem 3. We begin with proving part b). In Observation 14, we have seen that the competitive ratio of Blocking$_\delta$ is at least $\Omega(\log(n))$ if $\delta \in o(\log(n)/\log\log(n))$. By Lemma 15, we obtain the same lower bound for every δ in the range from $\Omega(\log(n))$ to $(n-4)/4$, and by Observation 16, we obtain the lower bound for δ at least $(n-4)/4$. Therefore, this proves the assertion of Theorem 3b. For part a), note that part b) implies that a competitive ratio of $o(\log(n))$ is only possible for δ in the range from $\Omega(\log(n)/\log\log(n))$ to $o(\log(n))$. Using Observation 14 in this range implies the assertion of Theorem 3a. ◀

3 Graph spanners in bounded genus graphs

In this section, we prove Theorem 4 about the existence of light spanners in bounded genus graphs. For this, we begin by introducing the greedy spanner.

3.1 The greedy spanner

The *greedy* $(1+\varepsilon)$*-spanner* was suggested by Althöfer et al. [2] and is formally defined as the output of Algorithm 2. After ordering the edges by weight, it iteratively adds edges if they are short in comparison to the distance of their endpoints in the graph constructed so far.

Algorithm 2 GreedySpanner($G = (V, E, w)$, ε).

sort $E = \{e_1, \dots, e_m\}$ such that $w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m)$
$H \leftarrow (V, \emptyset)$
for $i \leftarrow 1, \dots, m$ **do**
  if $d_H(u_i, v_i) > (1+\varepsilon)w(e_i)$, *where* $e_i = (u_i, v_i)$ **then**
    $H \leftarrow H \cup \{e_i\}$
return H

Note that the resulting graph H is indeed a $(1+\varepsilon)$-spanner of G. The output of the algorithm actually depends on the chosen order of the edges. In particular, when edge weights appear multiple times, there may be several possible outputs. However, this will not be important in our context. When we refer to *the* greedy spanner, we mean that we arbitrarily fix some output of the algorithm.

The greedy spanner fulfills the following two key properties: First, the algorithm implicitly executes Kruskal's algorithm for finding a minimum spanning tree, i.e., it adds all edges to H that Kruskal's algorithm adds. With this, we obtain the following.

▶ **Observation 17.** *The greedy spanner contains all edges of some minimum spanning tree of the input graph.*

The second key property, in fact, resembles the property of Blocking$_\delta$ in Lemma 7.

▶ **Observation 18** (Althöfer et al. [2])**.** *For every cycle C in the greedy spanner H and every edge e of C, we have $w(C \setminus \{e\}) > (1+\varepsilon)w(e)$. In other words, no proper subgraph of H is a $(1+\varepsilon)$-spanner of H.*

Proof. Let C be a cycle in the greedy spanner. Let $e = (u, v)$ be the edge in C that is added last. At the time it is added, we have $(1+\varepsilon)w(e) < d_H(u, v) \leq w(C \setminus \{e\})$ by definition of the algorithm. Since all other edges in C have lower or equal weight than e, the property is fulfilled for them as well. ◀

3.2 Spanners in planar graphs

Before investigating spanners in bounded genus graphs, we illustrate the technique for the special case of planar graphs, giving an alternate proof of the following result.

▶ **Theorem 19** (Althöfer et al. [2])**.** *For every planar graph G and $\varepsilon > 0$, the greedy $(1+\varepsilon)$-spanner of G has lightness at most $1 + \frac{2}{\varepsilon}$.*

Our proof uses similar ideas as in [31, Theorem 1] and is based on the following main idea: Fix an embedding of the greedy spanner in the plane and, in a suitable way, partition the greedy spanner into facial cycles, i.e., cycles that form the boundary of a face. Then use the fact that none of these cycles are short (cf. Observation 18).

▶ **Lemma 20.** *Let G be a planar graph, H be the greedy $(1+\varepsilon)$-spanner of G and* MST *be a minimum spanning tree of H. Fix an embedding of H in the plane. Then, we can associate with every edge $e \in H \setminus \mathrm{MST}$ a facial cycle C_e containing e, so that $C_e \neq C_{e'}$ for $e \neq e'$.*

Next, we illustrate how this can be combined with the fact that the greedy spanner does not contain short cycles (cf. Observation 18).

▶ **Lemma 21.** *Let G be a graph and H be the greedy $(1+\varepsilon)$-spanner of G. Let D be a subgraph of G such that we can associate with every edge $e \in H \setminus D$ a cycle C_e of H containing e, with the property that $\sum_{e \in H \setminus D} w(C_e) \leq 2w(H)$. Then, $w(H) \leq \left(1 + \frac{2}{\varepsilon}\right) w(D)$.*

Next, we show how this implies Theorem 19.

Proof of Theorem 19. Let G be a planar graph, let H be the greedy $(1+\varepsilon)$-spanner of G, and let MST denote a minimum spanning tree of H. By Observation 17, MST is also a minimum spanning tree of G, so that it suffices to show $w(H) \leq \left(1 + \frac{2}{\varepsilon}\right) w(\mathrm{MST})$. Since G is planar, its subgraph H is planar as well. Let us fix an embedding of H on the plane such that no two edges cross. By Lemma 20, there is a facial cycle C_e for every edge $e \in H \setminus \mathrm{MST}$ such that $C_e \neq C_{e'}$ for $e \neq e'$. As every edge of H is contained in at most two facial cycles, we have $\sum_{e \in H \setminus \mathrm{MST}} w(C_e) \leq 2w(H)$. Therefore, we can apply Lemma 21 with $D = \mathrm{MST}$ and obtain $w(H) \leq \left(1 + \frac{2}{\varepsilon}\right) w(\mathrm{MST})$. ◀

3.3 Generalization to bounded genus graphs

The *genus* of a graph G is the smallest integer g such that G can be embedded on an orientable surface of genus g. In this subsection, we study light spanners for the class of bounded genus graphs and prove Theorem 4. We begin by recalling the theorem.

▶ **Theorem 4** (restated). *For every $\varepsilon > 0$, the greedy $(1+\varepsilon)$-spanner of a graph of genus g has lightness at most $\left(1 + \frac{2}{\varepsilon}\right)\left(1 + \frac{2g}{1+\varepsilon}\right)$.*

Our proof is based on similar arguments as in [31, Theorem 2] and the main idea is roughly as follows: Given an embedding of the greedy spanner on a surface of genus g, first cut the surface along several edges such that we obtain a disk. Then, we can proceed along similar lines as for Theorem 19. In this work, we estimate more carefully the weight of the edges along which we cut so that we obtain a slightly improved bound than in [31]. We will use the following topological lemma for the first step.

▶ **Lemma 22.** *Let G be an unweighted connected graph of genus (exactly) $g \geq 1$. Fix an embedding of G on an orientable surface of genus g and let T be a spanning tree of G. Then, there is a subgraph D of G with $T \subseteq D$ and $|E(D) \setminus E(T)| \leq 2g$ such that, in the inherited embedding of D, there is only a single face and the edges in D bound a topological disk.*[3]

Proof. It is a standard fact from topology that, on a surface of genus g, one can embed precisely $2g$ closed curves that are non-separating, i.e., it is possible to draw $2g$ cycles on the surface such that cutting along all of them does not disconnect the surface. Every collection of $2g$ curves that are non-separating bounds a topological disk (see Figure 3).[4]

[3] A topological disk is a surface homeomorphic to a 2-dimensional disk. Intuitively, a topological disk is a continuous deformation of a 2-dimensional disk.

[4] This can be proven as follows: The Euler characteristic of a surface of genus g is $2-2g$ [27, Section 2.2] and cutting along a non-separating closed curve increases the Euler characteristic by 1.

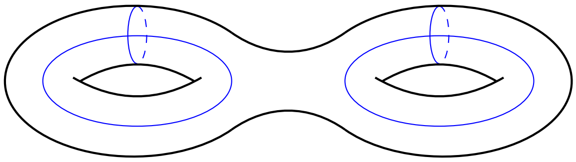

Figure 3 surface of genus 2 with 4 non-separating cycles bounding a topological disk.

We construct the set D greedily as follows (see Figure 4): Initially, let $D := T$. Ignoring all edges in $G \setminus D$, we have only a single face. Note that every edge in $G \setminus D$ closes a cycle with D. If we find an edge which only closes non-separating cycles, i.e, does not separate the surface into two faces, we add it to D. After this, the edges of D still only bound a single face. We repeat this step until we cannot find further edges whose addition would separate the surface into multiple faces.

Since there are at most $2g$ cycles on a surface of genus g that are non-separating, we have $|E(D) \setminus E(T)| \leq 2g$. It is left to show that D bounds a disk. By maximality of D, every edge $e \in G \setminus D$ is separating when added to D, i.e., in the inherited embedding of $D \cup \{e\}$, the edge e is incident to two faces. In particular, e is incident to two faces in the inherited embedding of every supergraph of D.

Consider again the embedding of the entire graph G. It is known from topological graph theory that a minimal genus embedding of a connected graph is cellular, i.e., every face of the embedding of G is a topological disk [39] (see [34, Proposition 3.4.1]). Since every edge $e \in G \setminus D$ is incident to two distinct faces, its removal merges the two corresponding disks along a connected part of their common boundary, which yields another disk. Iteratively removing all edges in $G \setminus D$ in this way, we thus obtain a cellular embedding of D. Since, by construction, D induces only a single face, we obtain that D bounds a topological disk.

For an illustration of the construction, consider Figure 4. In the example in the left column, the two green edges enclose non-separating cycles, whereas all blue edges close separating cycles. In the example in the right column, the half-dotted green edge in D could be replaced by the blue edge between u and v. ◀

Now, we have all the prerequisites in place to prove Theorem 4. The main idea is to give a similar construction as in Lemma 20 to partition the greedy spanner into facial cycles. Before delving into the proof, let us briefly comment on why Lemma 22 is not a reduction to the planar case, i.e., we cannot use the same construction as in Lemma 20.

Recall that the key ingredient of Lemma 20 was to define a partial order in which an edge e' precedes another edge e if e' is embedded on the inside of the cycle that e closes with MST. In the bounded genus case, if the cycle closed by e is non-separating, there is no such thing as "the inside" of the cycle. For example, consider the edge (u, v) in the right column of Figure 4 and the cycle it closes with MST. This cycle does not have an "inside" and cannot be decomposed into multiple faces. In particular, the cycle disappears after cutting the surface along D. However, it separates the disk bounded by D into two parts. Therefore, we have to consider cycles that include edges of $D \setminus \text{MST}$.

Proof of Theorem 4. Let G be some graph of genus g. Let H be the greedy $(1+\varepsilon)$-spanner of G and let MST denote a minimum spanning tree of H. By Observation 17, we know that MST is also a minimum spanning tree of G, so that it suffices to show $w(H) \leq \left(1 + \frac{2}{\varepsilon}\right)\left(1 + \frac{2g}{1+\varepsilon}\right) w(\text{MST})$.

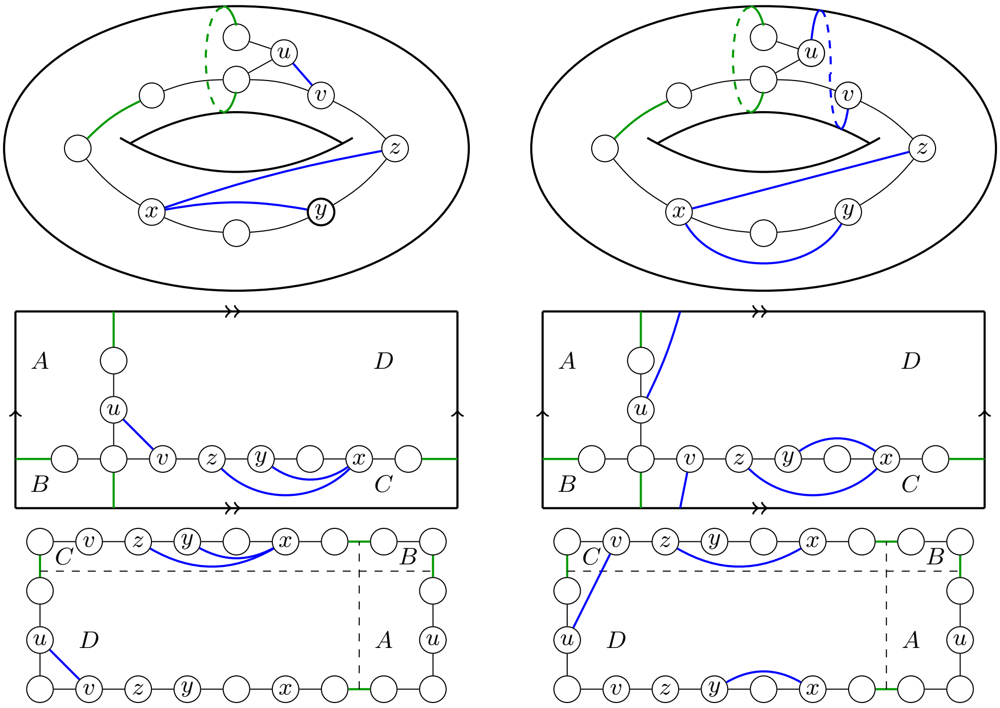

Figure 4 The two columns show the construction of D for the same graph with two different embeddings. The black edges belong to T, the green edges to $D \setminus T$, and the blue edges to $G \setminus D$. In each column, the first subfigure shows the embedding on the torus. The second subfigure shows a different representation: The torus is obtained by gluing together the opposite sides of the rectangle. The last subfigure shows the disk obtained by cutting the surface along D. Note that it contains every edge of D twice and therefore, every vertex up to 4 times. However, note that the embedding specifies between which copies of the vertices the blue edges have to be drawn. The capital letters A, B, C, D denote areas of the torus and are included for better orientation: Leaving area A to the left leads to area D, leaving A to the top leads to B and so on.

Let g' be the genus of H. If $g' = 0$, the assertion follows directly by Theorem 19. Therefore, we assume from now on $g' \geq 1$. Note that $g' \leq g$ because H is a subgraph of G. Fix an embedding of H on an orientable closed surface of genus g' such that no two edges cross. By Lemma 22, there is a subgraph D of H with $\mathrm{MST} \subseteq D$ such that

$$|E(D) \setminus E(\mathrm{MST})| \leq 2g' \leq 2g \tag{3}$$

and such that the edges of D induce only one face and bound a topological disk. Next, observe that, for every edge e in $H \setminus \mathrm{MST}$, we have $w(e) \leq w(\mathrm{MST})/(1+\varepsilon)$: Every edge e in $H \setminus \mathrm{MST}$ closes a cycle C together with the edges of MST. Using Observation 18, we obtain

$$w(e) < \frac{w(C \setminus \{e\})}{1+\varepsilon} \leq \frac{w(\mathrm{MST})}{1+\varepsilon}.$$

In particular, this is fulfilled for edges in $D \setminus \mathrm{MST}$. Combining this with (3), we obtain

$$w(D) \leq \left(1 + \frac{2g}{1+\varepsilon}\right) w(\mathrm{MST}). \tag{4}$$

The next step is to bound the weight of H by $(1 + 2/\varepsilon)w(D)$. For this, we use a similar construction as in Lemma 20 and show that it is possible to iteratively choose an edge e in $H \setminus D$ which, together with the edges of D and the edges chosen in previous iterations, closes a facial cycle C_e in the embedding of H.

In each iteration, we find a suitable edge as follows: Pick an arbitrary edge e of $H \setminus D$. If it defines a facial cycle together with D and edges chosen in previous iterations, we can simply choose e. Assume this is not the case. Note that e cuts the disk bounded by D in two parts and both contain edges in $H \setminus D$ to which no cycles have been assigned yet (otherwise e would close a suitable facial cycle). Pick the part whose boundary with D contains fewer edges (breaking ties arbitrarily) and pick a new edge e' in $H \setminus D$ which lies inside this half and has not yet been chosen in a previous iteration. Note that e' again cuts the disk in two parts and the boundary of the smaller part contains fewer edges of D than in the step before. Therefore, by repeating the steps above, we will end up with a suitable edge after finitely many steps. For example, on the left side of Figure 4, if we pick $e = (x, z)$, we will set $e' = (x, y)$ and this edge is suitable. After this, we can assign a facial cycle to (x, z) and then to (u, v). In the instance on the right, we can assign the cycles to the blue edges in any order.

Note that, in this construction, no two edges are assigned the same facial cycle. As every edge is contained in at most two facial cycles, we have

$$\sum_{e \in H \setminus D} w(C_e) \leq 2w(H). \tag{5}$$

Therefore, we can now apply Lemma 21 and obtain

$$w(H) \overset{\text{Lem 21}}{\leq} \left(1 + \frac{2}{\varepsilon}\right) w(D) \overset{(4)}{\leq} \left(1 + \frac{2}{\varepsilon}\right)\left(1 + \frac{2g}{1+\varepsilon}\right) w(\mathrm{MST}). \qquad \blacktriangleleft$$

Recall that Grigni showed that every graph of genus $g \geq 1$ contains a $(1 + \varepsilon)$-spanner of lightness at most $1 + (12g - 4)/\varepsilon$ [24]. Let us briefly comment on how our bound compares to Grigni's bound. For this, note that, for $g \geq 1$,

$$\left(1 + \frac{2}{\varepsilon}\right)\left(1 + \frac{2g}{1+\varepsilon}\right) = 1 + \frac{2}{\varepsilon} + \frac{2g}{1+\varepsilon} + \frac{4g}{\varepsilon(1+\varepsilon)} < 1 + \frac{2g}{\varepsilon} + \frac{2g}{\varepsilon} + \frac{4g}{\varepsilon} = 1 + \frac{8g}{\varepsilon}.$$

Therefore, our bound is stronger than Grigni's bound for every $g \geq 1$. Moreover, in the planar case (i.e., $g = 0$), we obtain a lightness of $1 + \frac{2}{\varepsilon}$. It was shown by Althöfer et al. [2, Theorem 5] that this is best possible, i.e., our bound is tight for planar graphs. Note that the worst-case lightness for spanners of graphs of genus g has to increase in g, since not every graph admits a light spanner. For example, for every $k \geq 3$ and almost all n, there is a graph on n vertices with girth at least k and at least $\frac{1}{4}n^{1+\frac{1}{k}}$ edges [32, Theorem 6.6].

4 Open problems

The key question in online graph exploration is whether the problem admits a constant-competitive algorithm [29]. While this problem remains open, our results suggest steps that might be needed towards a resolution of this question. Firstly, we have shown that the online graph exploration problem allows for a constant-competitive algorithm on graphs admitting a light spanner, in particular, minor-free graphs. This suggests that, for proving a non-constant general lower bound on the competitive ratio, one might require dense high-girth graphs or expanders [30]. Not even a competitive ratio of $o(\log(n))$ has yet been attained, and our results eliminate Blocking$_\delta$, for most values of δ, as a candidate for achieving this. It remains to close the gap between $\delta \in o(\log(n)/\log\log(n))$ and $\delta \in \Omega(\log(n))$.

Regarding spanners, we gave an improved upper bound on the lightness of spanners in bounded genus graphs. It is a natural question whether our bound is already tight for $g \geq 1$ or can further be improved. In particular, it is unclear whether the worst-case lightness for a fixed stretch must depend linearly on g.

References

1 Susanne Albers and Monika R. Henzinger. Exploring unknown environments. *SIAM Journal on Computing*, 29(4):1164–1188, 2000.

2 Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.

3 Sanjeev Arora, Michelangelo Grigni, David R. Karger, Philip N. Klein, and Andrzej Woloszyn. A polynomial-time approximation scheme for weighted planar graph TSP. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 33–41, 1998.

4 Baruch Awerbuch, Alan Baratz, and David Peleg. Cost-sensitive analysis of communication protocols. In *Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 177–187, 1990.

5 Alexander Birx, Yann Disser, Alexander V. Hopp, and Christina Karousatou. An improved lower bound for competitive graph exploration. *Theoretical Computer Science*, 868:65–86, 2021.

6 Antje Bjelde, Jan Hackfeld, Yann Disser, Christoph Hansknecht, Maarten Lipmann, Julie Meißner, Miriam Schlöter, Kevin Schewior, and Leen Stougie. Tight bounds for online TSP on the line. *ACM Transactions on Algorithms*, 17(1):3:1–3:58, 2021.

7 Vincenzo Bonifaci and Leen Stougie. Online k-server routing problems. *Theory of Computing Systems*, 45(3):470–485, 2009.

8 Glencora Borradaile, Erik D. Demaine, and Siamak Tazari. Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs. *Algorithmica*, 68(2):287–311, 2014.

9 Glencora Borradaile, Hung Le, and Christian Wulff-Nilsen. Minor-free graphs have light spanners. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 767–778, 2017.

10 Sebastian Brandt, Klaus-Tycho Foerster, Jonathan Maurer, and Roger Wattenhofer. Online graph exploration on a restricted graph class: Optimal solutions for tadpole graphs. *Theoretical Computer Science*, 839:176–185, 2020.

11 Shiri Chechik and Christian Wulff-Nilsen. Near-optimal light spanners. *ACM Transactions on Algorithms*, 14(3):1–15, 2018.

12 Erik D. Demaine, MohammadTaghi HajiAghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. *Combinatorica*, 30(5):533–552, 2010.

13 Xiaotie Deng and Christos H. Papadimitriou. Exploring an unknown graph. *Journal of Graph Theory*, 32(3):265–297, 1999.

14 Dariusz Dereniowski, Yann Disser, Adrian Kosowski, Dominik Paj, and Przemysław Uznański. Fast collaborative graph exploration. *Information and Computation*, 243:37–49, 2015.

15 Yann Disser, Jan Hackfeld, and Max Klimm. Tight bounds for undirected graph exploration with pebbles and multiple agents. *Journal of the ACM*, 66(6):40(41), 2019.

16 Yann Disser, Frank Mousset, Andreas Noever, Nemanja Skoric, and Angelika Steger. A general lower bound for collaborative tree exploration. *Theoretical Computer Science*, 811:70–78, 2020.

17 Stefan Dobrev, Rastislav Královič, and Euripides Markou. Online graph exploration with advice. In *Proceedings of the 19th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 267–278, 2012.

18 Franziska Eberle, Alexander Lindermayr, Nicole Megow, Lukas Nölke, and Jens Schlöter. Robustification of online graph exploration methods. In *Proceedings of the 36th Conference on Artificial Intelligence (AAAI)*, pages 9732–9740, 2022.

19 Paul Erdős. Extremal problems in graph theory. In *Proceedings of the Symposium on Theory of Graphs and its Applications*, pages 29–36, 1963.

20 Arnold Filtser and Shay Solomon. The greedy spanner is existentially optimal. *SIAM Journal on Computing*, 49(2):429–447, 2020.

21 Rudolf Fleischer and Gerhard Trippen. Exploring an unknown graph efficiently. In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA)*, pages 11–22, 2005.

22 Klaus-Tycho Foerster and Roger Wattenhofer. Lower and upper competitive bounds for online directed graph exploration. *Theoretical Computer Science*, 655:15–29, 2016.

23 Robin Fritsch. Online graph exploration on trees, unicyclic graphs and cactus graphs. *Information Processing Letters*, 168:106096, 2021.

24 Michelangelo Grigni. Approximate TSP in graphs with forbidden minors. In *Proceedings of the 27th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 1853, pages 869–877, 2000.

25 Michelangelo Grigni and Hao-Hsiang Hung. Light spanners in bounded pathwidth graphs. In *Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 467–477, 2012.

26 Michelangelo Grigni and Papa Sissokho. Light spanners and approximate TSP in weighted graphs with forbidden minors. In *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 852–857, 2002.

27 Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2000. URL: `https://cds.cern.ch/record/478079`.

28 Cor A.J. Hurkens and Gerhard J. Woeginger. On the nearest neighbor rule for the traveling salesman problem. *Operations Research Letters*, 32(1):1–4, 2004.

29 Bala Kalyanasundaram and Kirk R. Pruhs. Constructing competitive tours from local information. *Theoretical Computer Science*, 130(1):125–138, 1994.

30 Michael Krivelevich and Benjamin Sudakov. Minors in expanding graphs. *Geometric and Functional Analysis*, 19(1):294–331, 2009.

31 Nicole Megow, Kurt Mehlhorn, and Pascal Schweitzer. Online graph exploration: New results on old and new algorithms. *Theoretical Computer Science*, 463:62–72, 2012.

32 Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge University Press, 2017.

33 Shuichi Miyazaki, Naoyuki Morimoto, and Yasuo Okabe. The online graph exploration problem on restricted graphs. *IEICE transactions on information and systems*, 92(9):1620–1627, 2009.

34 Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins University Press, 2001.

35 David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of graph theory*, 13(1):99–116, 1989.

36 Daniel J. Rosenkrantz, Richard E. Stearns, and Philip M. Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing*, 6(3):563–581, 1977.

37 Daniel Russel and Leonidas J. Guibas. Exploring protein folding trajectories using geometric spanners. In *Pacific Symposium on Biocomputing (PSB)*, pages 42–53. World Scientific, 2005.

38 Bang Ye Wu, Kun-Mao Chao, and Chuan Yi Tang. Light graphs with small routing cost. *Networks*, 39(3):130–138, 2002.

39 John William Theodore Youngs. Minimal imbeddings and the genus of a graph. *Journal of Mathematics and Mechanics*, pages 303–315, 1963.