Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 30, no. 1, pp. 89–107 (2026)
DOI: 10.7155/jgaa.v30i1.2998

Stable and Dynamic Minimum Cuts

Mark de Berg [1] *Andrés López Martínez* [1] *Frits Spieksma* [1]

[1] Department of Mathematics and Computer Science
TU Eindhoven, The Netherlands

Submitted: Jan. 2025 Accepted: Sept. 2025 Published: March 2026

Article type: Regular paper Communicated by: R. Uehara

Abstract. We study four problems: maintaining an exact or ρ-approximate global min-cut, and an exact or ρ-approximate s-t cut, in dynamic graphs under the vertex-arrival model. We investigate the trade-off between the stability of a solution—the minimum number of *vertex flips* required to transform an induced bipartition into another when a new vertex arrives—and its quality. Trivially, in a graph with n vertices any cut can be maintained with $n/2$ vertex flips upon a vertex arrival. For all our problems, we obtain that this trivial stability bound is tight up to constant factors, even for a clairvoyant algorithm—one that knows the entire vertex-arrival sequence in advance. When ρ is large enough, we show that there are simple and stable algorithms for maintaining a ρ-approximate cut in both general and planar graphs. In view of the negative results, we also investigate the quality-stability trade-off in the amortized sense. For maintaining global and s-t minimum cuts, we show that the trivial $O(n)$ amortized stability bound is also tight up to constant factors. However, for maintaining a ρ-approximate cut, we show a lower bound of $\Omega(\frac{n}{\rho^2})$ average vertex flips, and give a (clairvoyant) algorithm with amortized stability $O\left(\frac{n \log n}{\rho \log \rho}\right)$. Moreover, for maintaining a ρ-approximate s-t cut in the clairvoyant setting, we establish a tight bound of $\Theta\left(\frac{\log n}{\log \rho}\right)$ average vertex flips.

1 Introduction

Given an undirected graph $G = (V, E)$, a *cut* $(S, \overline{S})$ is a partition of V into two non-empty sets S and $\overline{S}$. An s-t cut is a cut where $s \in S$ and $t \in \overline{S}$ (or vice versa), with s and t being two specified vertices in G. The *size* or *value* of a cut, denoted by $w(S, \overline{S})$, is the number of edges between S

Research supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 945045, and by the NWO Gravitation project NETWORKS under grant no. 024.002.003. An earlier version of this work appeared in WALCOM 2024 [3].

E-mail addresses: m.t.d.berg@tue.nl (Mark de Berg) a.lopez.martinez@tue.nl (Andrés López Martínez) f.c.r.spieksma@tue.nl (Frits Spieksma)

and $\overline{S}$. A *minimum s-t cut* (*s-t* min-cut) is an *s-t* cut with the smallest size. A *global minimum cut*, or simply *minimum cut* (min-cut), is the smallest *s-t* cut over all possible choices of s and t.

We refer to the problems of finding a global min-cut and an *s-t* min-cut as MINIMUM CUT and MINIMUM *s-t* CUT, respectively. Both are classical combinatorial optimization problems with a wide range of practical and theoretical applications [1]. Let $\lambda(G)$ and $\lambda(G,s,t)$ denote the values of a minimum cut and a minimum *s-t* cut in G, respectively. For a parameter $\rho \geq 1$, a *ρ-approximate cut* (resp. *ρ-approximate s-t cut*) $(X,\overline{X})$ is a cut (resp. *s-t* cut) with value at most $\rho\lambda(G)$ (resp. $\rho\lambda(G,s,t)$); that is, $w(X,\overline{X}) \leq \rho\lambda(G)$ (resp. $w(X,\overline{X}) \leq \rho\lambda(G,s,t)$). Throughout the paper, unless otherwise specified, the terms minimum cut and approximate cut refer to both global and *s-t* cuts.

We study four problems: maintaining an exact or approximate global min-cut, and an exact or approximate *s-t* cut, in the *vertex-arrival model*. In this model, the graph G changes over time as new vertices are added. Starting with an empty graph G_0, new vertices arrive one at a time, each accompanied by all their incident edges to previously arrived vertices. This process generates a sequence of graph instances $(G_0, G_1, \ldots, G_n)$, where n is the number of vertices in the final graph G_n. Each arrival corresponds to a *timestep*, with G_i denoting the state of the graph after the ith vertex and its edges have been added. We further assume that there is a timestep i^* after which each graph in the sequence is connected. For global min-cuts $i^* = 2$, and for *s-t* min-cuts i^* is the length of a *shortest s-t path* in G_n.

Traditionally, maintaining near-optimal solutions is the main objective in a dynamic setting. In practice, however, it may also be costly to implement the necessary changes to go from a valid solution at time i to a valid solution at time $i+1$. As a result, we are also interested in the *stability* of the maintained solutions or how *different* consecutive solutions are from each other. Following the framework by De Berg *et al.* [5], we say that a dynamic algorithm is a *γ-stable ρ-approximation algorithm* if, upon each vertex arrival, at most γ changes are required to transform the currently maintained solution into a solution in the augmented graph, and each solution is a ρ-approximation.

To define the difference between consecutive cuts in dynamic graphs, we use the notion of *vertex flips*. Let $(X,\overline{X})$ be a cut in a graph G, a *vertex flip* is the operation of a vertex v switching sides from X to $\overline{X}$ (or vice versa). Consider two consecutive graphs $G_i = (V,E)$ and $G_{i+1} = (V \cup \{v\}, E')$, and let $S_i = (X,\overline{X})$ and $S_{i+1} = (Y \cup \{v\}, \overline{Y} \setminus \{v\})$ be cuts in G_i and G_{i+1}, respectively. We say that the *difference* between S_i and S_{i+1} is the minimum number of vertex flips required to transform one cut into the other, and denote it by $D(S_i,S_{i+1}) = \min(\delta(S_i,S_{i+1}), |V| - \delta(S_i,S_{i+1}))$, where $\delta(S_i,S_{i+1}) = |X \cup Y| - |X \cap Y|$ is the cardinality of the symmetric difference $X \triangle Y$. Equivalently, we may write $D(S_i,S_{i+1}) = \min(|X \triangle Y|, |X \triangle \overline{Y}|)$. Note that this implies that $D(S_i,S_{i+1}) \leq \frac{i}{2}$. We remark that the newly arrived vertex v has no contribution to the calculation.

Related Work. In general, the challenge to maintain a (high-quality) solution to a dynamic problem while aiming to minimize changes to the solution, is known as optimization with bounded recourse. Here, the phrase "recourse" refers to the changes one is allowed to make to a solution. For various problems, results are known; we mention Gupta *et al.* [10] and Bernstein *et al.* [2] for work on maintaining matchings and flows, Imase and Waxman [14], Megow *et al.* [15] and Gu *et al.* [9] for work on maintaining (Steiner) trees and Hamiltonian cycles, and Feldkord *et al.* [6] and Han and Makino [11] for work on bin packing and knapsack. In many cases, the computational time spent in an iteration (the update time) is a relevant aspect of these works; in particular, for the min-cut problem results along these lines can be found in [7, 8, 13, 16, 17]. There is also work that focuses on the "difference" between two consecutive solutions, while not taking explicitly

Graph class	Type	Exact: Lower bound	Exact: Upper bound	ρ-Approximation: Lower bound	ρ-Approximation: Upper bound
General	Minimum Cut	$\frac{n-1}{2}$	$\frac{n-1}{2}$	$\frac{n-2}{2}$ for $\rho < \frac{n-2}{2} - 2$	$\frac{n-1}{2}$ for $\rho < \frac{n-2}{2} - 2$
				0 for $\rho > \frac{n-1}{2}$	2 for $\rho > \frac{n-1}{2}$
	Minimum s-t Cut	$\frac{n-3}{2}$	$\frac{n-1}{2}$	$\frac{n-2}{3}$ for $\rho < \sqrt{\frac{n-2}{3} + 1}$	$\frac{n-1}{2}$ for $\rho < \sqrt{\frac{n-2}{3} + 1}$
				$\frac{n-1}{4}$ for $\rho < \frac{n-1}{4} - 1$	$\frac{n-1}{2}$ for $\rho < \frac{n-1}{4} - 1$
Planar	Minimum Cut	$\frac{n-1}{2}$	$\frac{n-1}{2}$	$\frac{n-2}{2}$ for $\rho < 5$	$\frac{n-1}{2}$ for $\rho < 5$
				0 for $\rho \geq 5$	2 for $\rho \geq 5$
	Minimum s-t Cut	$\frac{n-3}{2}$	$\frac{n-1}{2}$	$\frac{n-2}{3}$ for $\rho < \frac{\sqrt{12n+129}-3}{6}$	$\frac{n-1}{3}$ for $\rho < \frac{\sqrt{12n+129}-3}{6}$
General (amortized)	Minimum Cut	$\frac{n}{16}$	$\frac{n-1}{4}$	$\Omega(n/\rho^2)$	$O\left(\frac{n \log n}{\rho \log \rho}\right)$
	Minimum s-t Cut	$\frac{n}{16}$	$\frac{n-1}{4}$	$\Omega\left(\frac{\log n}{\log \rho}\right)$	$O\left(\frac{\log n}{\log \rho}\right)$

Table 1: Combined summary of results on γ-stability for Minimum Cut and Minimum s-t Cut.

computational time into account; we mention Wasim and King [18] for work on Max-Cut, and De Berg *et al.* [4] for work on independent and dominating set. We follow this latter line of work, i.e., given the definition of difference between two min-cuts as formulated above, we establish trade-offs between the stability of a solution and its quality.

Our results. We study stable approximation algorithms for Minimum Cut and Minimum s-t Cut in the vertex-arrival model. More precisely, we obtain lower and upper bounds on the stability of dynamic min-cuts and s-t min-cuts on general graphs as well as planar graphs. The results are summarized in Table 1.

For general graphs, we show that maintaining a minimum cut may require up to $\frac{n-1}{2}$ vertex flips per iteration, while maintaining a minimum s-t cut may require up to $\frac{n-3}{2}$ vertex flips per iteration. These bounds are tight up to constant factors, as any transition between two cuts can be achieved with at most $\frac{n-1}{2}$ vertex flips. The results hold for both the *oblivious setting*—where the algorithm only knows the previously arrived elements of the vertex-arrival sequence—and the *clairvoyant setting*—where the algorithm has access to the entire sequence of vertex arrivals in advance. Similar results apply to the special case of planar graphs. In contrast, the problems become trivial in trees (where starting from a tree consisting of a single edge, we can always keep the partition of the vertex set induced by that edge as the cut) and in complete graphs (where we can always keep the same vertex as one of the parts of the cut). For general graphs in the amortized case, we show that in order to maintain a global minimum cut or minimum s-t cut, $\Theta(n)$ vertex flips are needed.

We now turn to maintaining ρ-approximate cuts. For general graphs, we show that similar to the exact case, maintaining a ρ-approximate cut may need $\frac{n-2}{2}$ vertex flips at some iteration, but only when $\rho < \frac{n-2}{2} - 2$. In contrast, when $\rho > \frac{n-1}{2}$, we show that two vertex flips per iteration suffices to maintain a ρ-approximate cut. Both results are tight up to constant terms. Similar results apply for planar graphs when $\rho < 5$ and $\rho \geq 5$, respectively. Finally, for general graphs in the amortized case, we show that to maintain a ρ-approximate cut at least $\Omega(n/\rho^2)$ vertex flips are needed. We accompany this result by giving a clairvoyant algorithm with amortized stability $O\left(\frac{n \log n}{\rho \log \rho}\right)$.

For the stability of maintaining ρ-approximate s-t cuts, we show that the trivial upper bound of $\frac{n-1}{2}$ vertex flips is tight up to constant factors. However, for clairvoyant algorithms, we obtain different lower bounds depending on the quality parameter ρ. Specifically, when $\rho = o(\sqrt{n})$, the lower bound is $\frac{n-2}{3}$, while for $\rho = \Omega(\sqrt{n})$, we establish a slightly weaker lower bound of $\frac{n-2}{3}$. For planar graphs, we derive a similar lower bound as in the case $\rho = o(\sqrt{n})$. Note that for oblivious algorithms the lower bound is $\frac{n-2}{3}$ in all cases. Finally, for general graphs in the amortized case, we establish a tight bound of $\Theta\left(\frac{\log n}{\log \rho}\right)$ on the stability of maintaining a ρ-approximate s-t cut. Note that this holds only for a clairvoyant algorithm.

Roadmap. Like Table 1, the presentation of the results is split into two parts. First, in Section 2, we present the results about maintaining global and s-t minimum cuts. Then, in Section 3, we discuss the results on maintaining ρ-approximate cuts and s-t cuts. We conclude with some general remarks in Section 4.

2 Maintaining Global and s-t Min-Cuts

2.1 Global Minimum Cuts

We start with the *oblivious* setting, in which an algorithm has no knowledge of the input sequence, other than the previously arrived vertices. We use $deg(v)$ to denote the degree of a vertex v.

Theorem 1. *There is no exact γ-stable algorithm for* Minimum Cut *in general graphs of size $n \geq 9$ for $\gamma < \left\lfloor \frac{n-1}{2} \right\rfloor$.*

Proof: For every $n \geq 9$, we present a sequence of graph instances $(G_1, \ldots, G_n)$ (see Figure 1) for which any exact algorithm requires at least $\left\lfloor \frac{n-1}{2} \right\rfloor$ vertex flips to maintain a minimum cut. Let G_{n-1} be the graph consisting of two cliques A and B, where $|A| = \left\lceil \frac{n-1}{2} \right\rceil$ and $|B| = \left\lfloor \frac{n-1}{2} \right\rfloor$, connected to each other by means of two edges (a_1, b_1) and (a_2, b_2) for arbitrary $a_1, a_2 \in A$ and $b_1, b_2 \in B$. The graph G_n has one more vertex u, which is connected by a single edge to an arbitrary vertex in clique A.

Note that for $n-1 \geq 8$ the cliques A and B have at least four vertices, and so the only minimum cut for G_{n-1} is (A, B) which has value 2. When the vertex u arrives, $(\{u\}, A \cup B)$ is the unique minimum cut. Hence, any algorithm maintaining a minimum cut must move all $\left\lfloor \frac{n-1}{2} \right\rfloor$ vertices of B into the part of the cut containing A. □

At first glance, it might appear that an algorithm that can react to an incoming update only with past information is too restrictive when trying to obtain a good quality-stability trade-off.

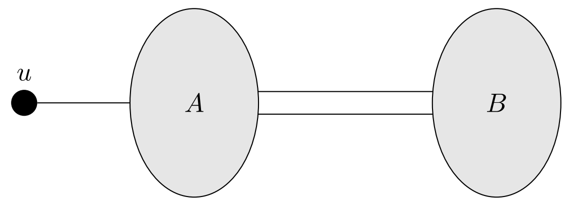

Figure 1: Graph G_n for the proof of the lower bound in Theorem 1. Cliques A and B (gray) are connected by means of two edges between arbitrary vertices. Vertex u (black) is connected to an arbitrary vertex in clique A. After its arrival, any algorithm must perform $\frac{n-1}{2}$ vertex flips to maintain a minimum cut.

However, even in a setting where the algorithm has access to the entire vertex-arrival sequence in advance—what we call, the *clairvoyant* setting—Theorem 1 still holds. Simply observe that, in the proof of Theorem 1, the algorithm must find solutions $S_{n-1} = (A, B)$ and $S_n = (\{u\}, A \cup B)$ at timesteps $n-1$ and n respectively since these are the only available min-cuts in G_{n-1} and G_n, respectively.

Corollary 1. *There is no clairvoyant exact γ-stable algorithm for* Minimum Cut *in general graphs of size $n \geq 9$ for $\gamma < \left\lfloor \frac{n-1}{2} \right\rfloor$.*

In search of better quality-stability trade-offs, we turn our attention to planar graphs. But like the general case, we obtain that $\Omega(n)$ vertex flips may be needed. This is tight with respect to the trivial upper bound of $\frac{n-1}{2}$ vertex flips.

Corollary 2. *There is no exact γ-stable algorithm for* Minimum Cut *in planar graphs for $\gamma < \left\lfloor \frac{n-1}{2} \right\rfloor$, even in the clairvoyant setting.*

Proof: Similar to the proof of Theorem 1, but replacing the cliques A and B by two planar graphs of connectivity at least 3 (e.g., each a maximal planar graph on $\frac{n-1}{2}$ vertices). □

2.2 Minimum s-t Cuts

Next, we consider the problem of maintaining a minimum s-t cut. Unlike global cuts, s-t cuts specifically require that two designated vertices, s and t, reside on opposite sides of the bipartition. This distinction provides additional structure to the problem. For example, an s-t cut $(X, \overline{X})$ with $|X| = 1$ means that $X = \{s\}$ or $X = \{t\}$. Consequently, the counterexample used in Theorem 1 does not directly apply to minimum s-t cuts. However, despite the added structure, we show that stability in the exact case remains challenging, as $\Omega(n)$ vertex flips may still be required.

Theorem 2. *There is no exact γ-stable algorithm for* Minimum s-t Cut *in general graphs of size $n \geq 7$ for $\gamma < \left\lfloor \frac{n-3}{2} \right\rfloor$.*

Proof: For every $n \geq 7$, we present a sequence of instances $(G_1, \ldots, G_n)$ for which any exact stable algorithm requires at least $\left\lfloor \frac{n-3}{2} \right\rfloor$ vertex flips to maintain a minimum s-t cut. The graph G_n features the two designated vertices s and t, connected by a single edge, along with two cliques A and B of roughly equal size, with $|A| \geq \left\lfloor \frac{n-3}{2} \right\rfloor$ and $|B| \geq \left\lfloor \frac{n-3}{2} \right\rfloor$. Vertex s is connected to one arbitrary vertex in A, while t is connected to two arbitrary vertices in B. Additionally, the graph includes a vertex u that is connected with two arbitrary vertices in A and to every vertex in B. See Figure 2 for an illustration.

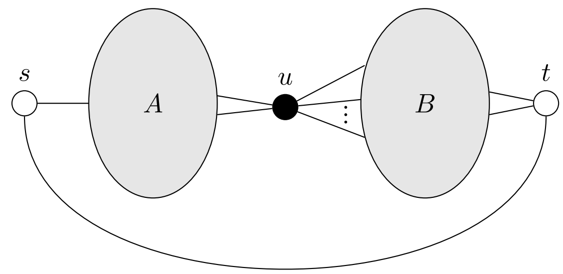

Figure 2: Graph G_n for the proof of the lower bound in Theorem 2.

Consider the vertex arrival sequence where s and t are revealed first, followed by the vertices in $A \cup B$, and finally vertex u. We now argue that, after the final vertex u is revealed, any algorithm maintaining a minimum s-t cut must perform $\lfloor \frac{n-3}{2} \rfloor$ vertex flips. To see this, first observe that, in G_{n-1}, the unique minimum s-t cut is $X = (\{s\} \cup A, \{t\} \cup B)$, with a value of 1. However, when vertex u is added, the maintained cut X is no longer optimal, as its value increases to at least 3 (depending on which side of the bipartition u is placed). At this point, there is a unique s-t cut of minimum value 2; namely, $Y = (\{s\}, A \cup B \cup \{u, t\})$. To transition to the new optimal cut Y at timestep n, the algorithm must then perform $\lfloor \frac{n-3}{2} \rfloor$ vertex flips, since $D(X, Y) = |A|$. □

Similarly to global minimum cuts, the stability of maintaining a minimum s-t cut is equally challenging in planar graphs. The following is an immediate corollary of Theorem 2 when replacing A and B with two maximal planar graphs.

Corollary 3. *There is no exact γ-stable algorithm for* Minimum s-t Cut *in planar graphs for $\gamma < \lfloor \frac{n-3}{2} \rfloor$, even in the clairvoyant setting.*

2.3 Amortized Analysis

2.3.1 Global Minimum Cuts

We saw in Theorem 1 that there are vertex arrival sequences where at least one iteration requires $\frac{n-1}{2}$ vertex flips to maintain an exact min-cut. It is natural to ask whether this behavior is only limited to a handful of iterations. If true, we could design an algorithm that, on average, requires only a few vertex flips per iteration. However, as the following shows, the average, or amortized stability—denoted by $\overline{\gamma}$—of maintaining an exact min-cut is not much better than the worst case: there exists a sequence of vertex arrivals such that each arrival induces $\Omega(n)$ many vertex flips.

Theorem 3. *There is no exact algorithm for* Minimum Cut *in general graphs with amortized stability $\overline{\gamma} < \frac{n}{16} - 2$.*

Proof: For every $n \geq 32$, we present a sequence of graph instances $(G_1, \dots, G_n)$ for which any exact stable algorithm requires at least $\frac{n}{16} - 2$ vertex flips *on average* (hence $\Omega(n^2)$ flips *in total*) to maintain a minimum cut. The graph G_n consists of three cliques A, B, and C of equal size $\frac{n+5}{4}$, connected to each other by the edges (a_1, b_1), (a_2, c_1), and (b_2, c_2), for arbitrary $a_1, a_2 \in A$, $b_1, b_2 \in B$, and $c_1, c_2 \in C$. Additionally, there is one more edge (a_3, c_3) for arbitrary $a_3 \in A$ and $c_3 \in C$, and two more edges (b_3, c_4), (b_4, c_5) for arbitrary $b_3, b_4 \in B$ and $c_4, c_5 \in C$. Graph G_n

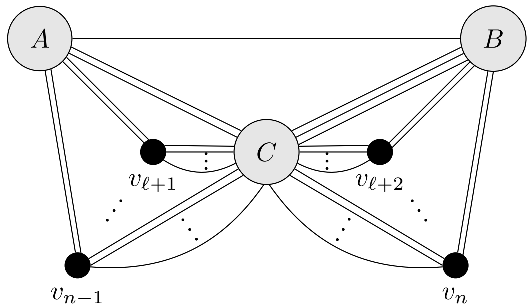

Figure 3: Graph G_n for the proof of the lower bound in Theorem 3. Cliques A, B, and C are highlighted in gray. The vertices in set D are highlighted in black. Of these vertices, those to the left (resp. right) of clique C belong to the set D_1 (resp. D_2). The alternated arrival of vertices from D_1 and D_2 induce $\Omega(n)$ vertex flips per iteration.

has $\frac{n-15}{4}$ additional vertices, denoted by set D, each of which shares an edge with every vertex in clique C. Moreover, D is partitioned into two disjoint sets D_1 and D_2 of equal size $|D|/2$, where each vertex in D_1 (resp. D_2) is connected to exactly two arbitrary vertices in clique A (resp. B). There are no more edges in G_n. See Figure 3 for an illustration.

Consider the vertex arrival sequence $\sigma = (v_1, v_2, \ldots, v_n)$ where the vertices in $A \cup B \cup C$ all arrive in the prefix subsequence $\sigma_1 = (v_1, v_2, \ldots, v_\ell)$, with $\ell = \frac{3(n+5)}{4}$, and vertices in D arrive according to subsequence $\sigma_2 = (v_{\ell+1}, \ldots, v_n)$. Let σ_2 be a permutation of vertices in D such that $v_{\ell+i} \in D_1$ if i is odd, and $v_{\ell+i} \in D_2$ if i is even. We prove the main claim by showing that each vertex arrival in σ_2 induces $\frac{n+5}{4}$ vertex flips. The idea is to have the minimum cut *oscillate* between cuts $(A, \overline{A})$ and $(B, \overline{B})$ as vertices in D_1 and D_2 arrive alternately.

First, we notice that after the ℓ-th vertex has arrived, cuts $(A, \overline{A})$, $(B, \overline{B})$ and $(C, \overline{C})$ have values 3, 4 and 5, respectively. Any other cut in the subgraph $G[A \cup B \cup C]$ must cross a clique and thus have value at least $\frac{n+1}{4}$. Thus, the min-cut at timestep ℓ is $S_\ell = (A, \overline{A})$. Next, after vertex $v_{\ell+1} \in D_1$ arrives, the value of cuts $(A, \overline{A})$ and $(C, \overline{C})$ increase by 2 and $\frac{n+5}{4}$ units, respectively; while the value of cut $(B, \overline{B})$ remains unchanged. Therefore, the min-cut at timestep $\ell+1$ becomes $S_{\ell+1} = (B, \overline{B})$. Similarly, after vertex $v_{\ell+2} \in D_2$ arrives, the value of cut $(B, \overline{B})$ increases by 2 units while cut $(A, \overline{A})$ remains unchanged, thus making $S_{\ell+2} = (A, \overline{A})$ the min-cut again; and so on for the remaining vertex arrivals in σ_2. The key observations are (i) that the min-cut at every timestep i is unique and has value less than $\frac{n+5}{4} - 1$, and (ii) $|w(A, \overline{A}) - w(B, \overline{B})| = 1$ is an invariant throughout the arrival sequence σ_2. From observation (ii), it follows that every vertex arrival in σ_2 increases the connectivity of the graph in a single unit. Now, by definition of G_n, we know that $\delta_{A,B} = D((A, \overline{A}), (B, \overline{B})) = \frac{n+5}{4}$. So, the total number of vertex flips performed for sequence σ_2 is $|\sigma_2| \cdot \delta_{A,B} = \frac{n-15}{4} \cdot \frac{n+5}{4}$. And averaged over the entire sequence σ, we obtain an amortized stability of at least[1] $\frac{n}{16} - 2$; which proves the theorem. □

Like Theorem 1, the lower bound of Theorem 3 is tight up to constant factors. To see this, simply consider a vertex arrival sequence where each update induces the maximum number of vertex flips at each iteration. Clearly, the amortized stability in this case is $\frac{n-1}{4}$.

[1] Because $\frac{n}{16} - 2 < \frac{n-15}{4} \cdot \frac{n+5}{4} \cdot \frac{1}{n} = \frac{n-10}{16} - \frac{75}{16 \cdot n}$ for any $n > 5$.

2.3.2 Minimum s-t Cuts

We conclude this section by noting that the construction used in the proof of Theorem 3 is also applicable to MINIMUM s-t CUT. Simply, let s be a vertex of clique A and t a vertex of clique B, with the single edge (s,t) connecting cliques A and B.

Corollary 4. *There is no exact algorithm for* MINIMUM *s-t* CUT *in general graphs with amortized stability $\overline{\gamma} < \frac{n}{16} - 2$.*

3 Maintaining Approximate Global and s-t Cuts

3.1 Approximate Global Cuts

We now consider the stability of maintaining approximate cuts. Theorem 1 shows that maintaining an exact solution is very expensive in terms of stability. Perhaps surprisingly, the following result shows that no better trade-off can be achieved for approximate solutions.

Theorem 4. *There is no γ-stable ρ-approximation algorithm for* MINIMUM CUT *in general graphs of size $n \geq 10$ for $\rho < \left\lfloor \frac{n-2}{2} \right\rfloor - 2$ and $\gamma < \left\lfloor \frac{n-2}{2} \right\rfloor$.*

Proof: For every $n \geq 10$, we present a sequence of graph instances $(G_1, \ldots, G_n)$ for which any approximation algorithm requires at least $\left\lfloor \frac{n-2}{2} \right\rfloor$ vertex flips to obtain an approximation ratio less than ℓ where $1 < \ell \leq \left\lfloor \frac{n-2}{2} \right\rfloor - 2$ (when $\ell = \left\lfloor \frac{n-2}{2} \right\rfloor - 2$ the theorem follows)[2]. The graph G_n has two cliques A and B of roughly equal size such that $|A| \geq \left\lfloor \frac{n-2}{2} \right\rfloor$ and $|B| \geq \left\lfloor \frac{n-2}{2} \right\rfloor$, connected to each other by means of a single edge (a,b), for arbitrary $a \in A$ and $b \in B$. In addition, G_n has two more vertices u and w. Vertex w has $deg(w) = 2(\ell - 1)$ and shares half of its edges with arbitrary vertices from clique A and the other half with arbitrary vertices from clique B. Vertex u has $deg(u) = 1$ and is connected to an arbitrary vertex in clique A. See Figure 4 for an illustration of graph G_n.

Consider any dynamic algorithm for maintaining a ρ-approximate cut and let S_i denote the cut maintained by the algorithm after the first i vertices have arrived. Consider the graph defined above, where the vertices in $A \cup B$ arrive in the first $n-2$ timesteps, followed by vertex u at timestep $n-1$ and w at timestep n. First, we show that at timestep $n-2$—that is, right after the vertices in $A \cup B$ have arrived—the algorithm must maintain the cut (A, B) as the solution; *i.e.*, $S_{n-2} = (A, B)$. This follows from the fact that the graph G_{n-2} has a single min-cut of value 1—namely, the cut (A, B)—and any other cut in G_{n-2} has value at least $\left\lfloor \frac{n-2}{2} \right\rfloor - 1 > \ell$. Hence, only the cut (A, B) has approximation ratio less than ℓ.

We now show that at timestep n—after vertices u and w arrive—our algorithm will have performed $\left\lfloor \frac{n-2}{2} \right\rfloor$-many vertex flips. First, we observe that at timestep $n-1$ (after vertex u arrives) the graph G_{n-1} presents only three ℓ-approximate cuts: the two min-cuts $(A \cup \{u\}, B)$ and $(\{u\}, A \cup B)$, and the 2-approximate cut $(A, B \cup \{u\})$. These are the only ℓ-approximate cuts because any other cut partitions clique A and/or clique B into two non-empty sets, thus cutting at least $\left\lfloor \frac{n-2}{2} \right\rfloor - 1$ edges. Thus, at timestep $n-1$, our algorithm must pick one of these cuts as S_{n-1}. Next, after the final vertex w arrives at timestep n, only the (unique) min-cut $(\{u\}, A \cup B \cup \{w\})$ is a valid ρ-approximate cut[3]. Therefore, at timestep n, our algorithm must

[2] Solving $1 < \left\lfloor \frac{n-2}{2} \right\rfloor - 2$ for integer n results in our stated bound of $n \geq 10$.

[3] Because any clique-crossing cut has value at least $\left\lfloor \frac{n-2}{2} \right\rfloor - 1 > \ell$ since $deg(v) \geq \left\lfloor \frac{n-2}{2} \right\rfloor - 1 \; \forall v \in A \cup B$. And any non-clique-crossing cut (except the min-cut) must cut at least ℓ edges: one edge shared by cliques and $\ell - 1$ edges shared by w with one of the cliques.

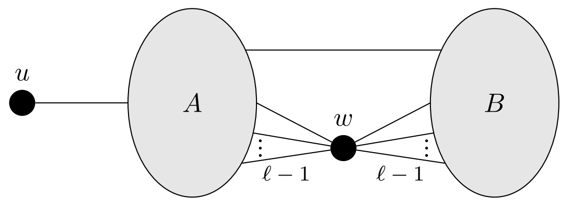

Figure 4: Graph G_n for the proof of the lower bound in Theorem 4. Cliques A and B (gray) are connected by a single edge between arbitrary vertices. Vertex u (black) is connected to an arbitrary vertex in clique A, and vertex w (black) to $\ell - 1$ arbitrary vertices in A and B, respectively. After the arrival of both u and w, any dynamic algorithm must perform $\frac{n-2}{2}$ vertex flips to maintain a ρ-approximate cut.

find $S_n = (\{u\}, A \cup B \cup \{w\})$. Now we show that no matter the choice for S_{n-1}, there is a timestep where the difference between two consecutive solutions is $\lfloor \frac{n-2}{2} \rfloor$.

Case 1. Let $S_{n-1} = (A \cup \{u\}, B)$. (The case for $S_{n-1} = (A, B \cup \{u\})$ is similar and is thus omitted.) As mentioned above, the only valid solution at timestep n is $S_n = (\{u\}, A \cup B \cup \{w\})$, but $D(S_{n-1}, S_n) \geq \lfloor \frac{n-2}{2} \rfloor$; that is, cut S_n is at least $\lfloor \frac{n-2}{2} \rfloor$ vertex flips away from S_{n-1}. Therefore, at least $\lfloor \frac{n-2}{2} \rfloor$ vertex flips are needed at timestep n.

Case 2. Let $S_{n-1} = (\{u\}, A \cup B)$. In contrast to the previous case, the difference between consecutive solutions S_{n-1} and S_n here is $D(S_{n-1}, S_n) = 0$. However, the difference between S_{n-2} and S_{n-1} is $D(S_{n-2}, S_{n-1}) \geq \lfloor \frac{n-2}{2} \rfloor$, because $S_{n-2} = (A, B)$. Therefore, at least $\lfloor \frac{n-2}{2} \rfloor$ vertex flips are performed at timestep $n - 1$.

This proves that any algorithm on $(G_1, \ldots, G_n)$ requires at least $\lfloor \frac{n-2}{2} \rfloor$ vertex flips to find an ρ-approximate cut such that $\rho < \ell$. Since $\ell = \lfloor \frac{n-2}{2} \rfloor - 2$ in the worst case, the claim follows. □

Similar to the case of maintaining a minimum cut, a clairvoyant algorithm fares no better than an oblivious one.

Corollary 5. *There is no clairvoyant γ-stable ρ-approximation algorithm for* Minimum Cut *in general graphs of size $n \geq 10$ for $\rho < \lfloor \frac{n-2}{2} \rfloor - 2$ and $\gamma < \lfloor \frac{n-2}{2} \rfloor$.*

Proof: This follows directly from the proof of Theorem 4. Observe that at timesteps $n - 2$ and n, respectively, the space of valid ρ-approximate cuts contains a single solution. Hence, even a clairvoyant algorithm is required to find solutions $S_{n-2} = (A, B)$ and $S_n = (\{u\}, A \cup B)$ at timesteps $n - 2$ and n respectively. The only freedom that the algorithm can exert is at timestep $n - 1$, where the space of valid ρ-approximate cuts contains three possible solutions. But as we have proved, any of the three possibilities for S_{n-1} still lead the algorithm to make $\lfloor \frac{n-2}{2} \rfloor$ vertex flips in some timestep. □

Similarly, we have the following for planar graphs.

Corollary 6. *There is no γ-stable ρ-approximation algorithm for* Minimum Cut *in planar graphs for $\rho < 5$ and $\gamma < \frac{n-2}{2}$, even in the clairvoyant setting.*

Proof: For every $n \geq 24$ divisible by 12, there is a planar graph H on n vertices with edge-connectivity five, and with at least six vertices incident to the *outer face* (see e.g. [12, Fig. 1]).

Then, we can use the proof of Theorem 4 by replacing the clique clusters A and B with two copies of H on $\frac{n-2}{2}$ vertices (assuming that $\frac{n-2}{2}$ is divisible by 12) and setting $1 < \ell \leq 4$. Notice that, since the planar graph H has more than four vertices incident to the outer face, the arrived vertex w can indeed share at most one edge with each of these vertices while the overall graph remains planar. □

3.2 Approximate s-t Cuts

We now investigate the stability of approximating MINIMUM s-t CUT. Similar to the exact case, we show that maintaining a ρ-approximate s-t cut may require $\Omega(n)$ many vertex flips, even for a clairvoyant algorithm. We first establish the lower bounds for clairvoyant algorithms, and then derive the bound for oblivious algorithms as a corollary.

Theorem 5. *There is no clairvoyant γ-stable ρ-approximation algorithm for* MINIMUM s-t CUT *in general graphs of size $n \geq 13$ for $\rho < \lfloor \frac{n-1}{4} \rfloor - 1$ and $\gamma < \lfloor \frac{n-1}{4} \rfloor$.*

Proof: For every $n \geq 13$ we present a sequence of graph instances $(G_1, \dots, G_n)$ for which any dynamic approximation algorithm requires at least $\lfloor \frac{n-1-\ell}{3} \rfloor$ vertex flips to obtain an approximation ratio $\rho < \ell$, with $2 \leq \ell \leq \lfloor \frac{n-1}{4} \rfloor$. (The theorem follows when $\ell = \lfloor \frac{n-1}{4} \rfloor$.) The graph G_n consists of the two vertices s and t, as well as three cliques A, B, and C, each of size at least $\lfloor \frac{n-1-\ell}{3} \rfloor$. Vertex s belongs to clique A and is connected to an arbitrary vertex in B, while t belongs to C, and is connected to an arbitrary vertex in B. The graph also features a vertex u that shares $\ell - 1$ edges with cliques A and B, respectively, as well as a set D of ℓ additional vertices. Each vertex in D is connected to ℓ vertices in B and C, respectively, with the exception of one vertex $w \in D$, which shares $\ell - 1$ edges with B and C, respectively. See Figure 5 for an illustration of G_n.

Consider the vertex arrival sequence where the vertices in $A \cup B \cup C$ are revealed during the first $k = |A| + |B| + |C|$ timesteps, followed by vertex u, then w, and finally the rest of the vertices in D. We show that after the last vertex of D is revealed, any algorithm, even if clairvoyant, must perform $\lfloor \frac{n-1}{4} \rfloor$ vertex flips to maintain a ρ-approximate s-t cut, where $\rho < \ell$.

First, at timestep k, there are exactly two ρ-approximate s-t cuts: $X_k = (A, B \cup C)$ and $Y_k = (A \cup B, C)$, both of which are minimum s-t cuts. (Note that any other s-t cut would partition one of the three cliques, resulting in a cut size of at least $\lfloor \frac{n-1-\ell}{3} \rfloor - 1 > \rho$.) Hence, our algorithm must maintain one of X_k or Y_k at timestep k.

We analyze the two cases. At timestep $k+1$, when vertex u is revealed, the cut $Y_{k+1} = (A \cup B \cup \{u\}, C)$ becomes the only feasible ρ-approximate s-t cut. This is because the alternative cuts $(A \cup \{u\}, B \cup C)$ and $(A, \{u\} \cup B \cup C)$ both have sizes $\ell > \rho$. Consequently, if the algorithm maintained X_{n-2} at timestep k, it would be forced to transition to Y_{k+1} at timestep $k+1$, already incurring in $\lfloor \frac{n-1-\ell}{3} \rfloor$ vertex flips. On the other hand, if the algorithm maintained Y_k at timestep k, it could transition to Y_{k+1} without any vertex flips, as $D(Y_k, Y_{k+1}) = 0$.

Assuming the algorithm maintained cuts Y_{k+1} and Y_k at timesteps $k+1$ and k, respectively, we consider what happens at timesteps $i \in [k+2, n-1]$. During this interval, the minimum s-t cut value is ℓ, given by the two cuts $X_1 = (A, \overline{A})$ and $X_2 = (A \cup \{u\}, \overline{A \cup \{u\}})$. Let D_i denote the set of vertices from D revealed after timestep i. During interval $[k+2, n-1]$, the s-t cuts $Y_1 = (C, \overline{C})$ and $Y_2 = (C \cup D_i, \overline{C \cup D_i})$ are the only feasible ρ-approximate s-t cuts, both of which are $\lfloor \frac{n-1-\ell}{3} \rfloor$ vertex flips away from either X_1 or X_2. Hence, either one of Y_1 or Y_2 must be maintained by the algorithm during timesteps $i \in [k+2, n-1]$. (Note that it takes 0 vertex flips to transition from Y_{k+1} to either of Y_1 or Y_2 at timestep $k+2$.)

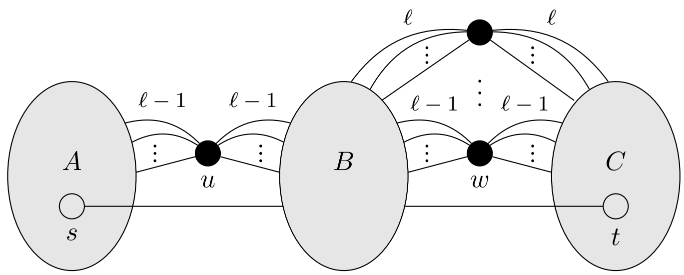

Figure 5: Graph G_n for the proof of the lower bound in Theorem 5.

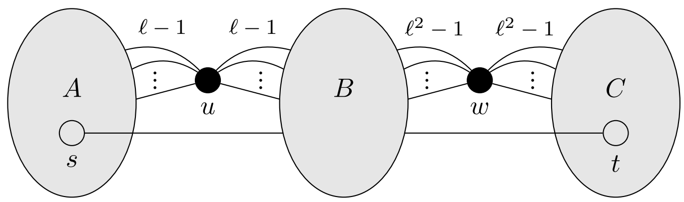

Figure 6: Graph G_n for the proof of the lower bound in Corollary 7.

At timestep n, when the last vertex from D is revealed, the only feasible ρ-approximate s-t cuts are now X_1 or X_2. This is because the cost of Y_1 (resp. Y_2) is ℓ^2, while the cost of X_1 (resp. X_2) is ℓ, and we wish for an approximation ratio $\rho < \ell$. Therefore, at timestep n, the algorithm must transition to X_1 or X_2, incurring in $\left\lfloor \frac{n-1-\ell}{3} \right\rfloor$ vertex flips.

Therefore, whether the flips occur at timestep $k+1$ when u is revealed, or at timestep n when the last vertex of set D is revealed, the algorithm must perform at least $\left\lfloor \frac{n-1-\ell}{3} \right\rfloor$ vertex flips. $\square$

When $\rho = o(\sqrt{n})$, we can obtain a slighlty improved lower bound.

Corollary 7. *There is no clairvoyant γ-stable ρ-approximation algorithm for* Minimum *s-t* Cut *in general graphs of size $n \geq 11$ for $\rho < \sqrt{\left\lfloor \frac{n-2}{3} \right\rfloor + 1}$ and $\gamma < \left\lfloor \frac{n-2}{3} \right\rfloor$.*

Proof: We follow the same construction of G_n as in the proof of Theorem 5, with the difference that the set D now consists of a single vertex w, which, instead of having degree $2(\ell-1)$, shares $\ell^2 - 1$ edges with cliques B and C, respectively. As a result, each of the three cliques has a size of at least $\left\lfloor \frac{n-2}{3} \right\rfloor$. The maximum value of ℓ is then $\ell = \sqrt{\left\lfloor \frac{n-2}{3} \right\rfloor + 1}$. See Figure 6 for an illustration of G_n.

Similar to the original proof, we consider the vertex arrival sequence where the vertices in $A \cup B \cup C$ are revealed during the first $n-2$ timesteps, followed by the vertices u and w at timesteps $n-1$ and n, respectively. Proceeding analogously to the remainder of the proof of Theorem 5, it follows that after vertex w is revealed, a clairvoyant algorithm must perform $\left\lfloor \frac{n-2}{3} \right\rfloor$ vertex flips to maintain a ρ-approximate s-t cut, where $\rho < \ell$. $\square$

Corollary 8. *There is no oblivious γ-stable ρ-approximation algorithm for* Minimum *s-t* Cut *in general graphs of size $n \geq 10$ for $\rho < \left\lfloor \frac{n-1}{3} \right\rfloor$ and $\gamma < \frac{n-1}{3}$.*

Proof: We follow the same construction as in the proof of Corollary 7, with the exception of the final timestep. Specifically, we consider the vertex arrival sequence where $A \cup B \cup C$ are revealed during the first $n-1$ timesteps, and u is revealed at timestep n. Since our algorithm is oblivious, in the worst case, at timestep $n-1$, the algorithm maintains the cut $X_{n-1} = (A, B \cup C)$. When u is revealed, the cut $Y_n = (A \cup B \cup u, C)$ becomes the only feasible ρ-approximate s-t cut. Since $D(X_{n-1}, Y_n) \geq \lfloor \frac{n-1}{3} \rfloor$, the claim follows. □

For planar graphs, we get similar results for clairvoyant and oblivious algorithms.

Theorem 6. *There is no clairvoyant γ-stable ρ-approximation algorithm for* Minimum s-t Cut *in planar graphs for $\rho < \frac{\sqrt{12n+129}-3}{6}$ and $\gamma < \lfloor \frac{n-2}{3} \rfloor$.*

Proof: We follow a similar approach to the proof of Corollary 7, with the key difference that the sets A, B, and C are no longer cliques but star graphs. Specifically, A is a star graph with s as the central vertex, C is a star graph with t at the center, and B is a star graph centered at a vertex v, where v is connected by an edge to both s and t. As before, we consider the vertex arrival sequence where $A \cup B \cup C$ are revealed first, followed by vertices u and w in that order. Here, u shares $\ell - 1$ edges with the stars A and B, respectively, while w shares $\ell^2 - 1$ edges with the stars B and C, respectively.

The rest of the proof proceeds in the same way as in Corollary 7 (and thus Theorem 5), but the analysis of the approximation bound deserves special mention. Since the graph must remain planar, the edges connecting u and w to vertices in star B cannot cross. Yet, if u and w shared three or more endpoints in B, the graph would include $K_{3,3}$ as a minor, violating planarity. Thus, they can share at most two endpoints in B. In the worst case, every vertex of B is connected to at least one of u or w, which implies $\frac{n-2}{3} = \ell^2 + \ell - 4$. Solving for ℓ, we find $\ell = \frac{\sqrt{12n+129}-3}{6}$. Since we are interested in an approximation ratio $\rho < \ell$, the theorem follows. □

Similar to Corollary 8, the following is an immediate corollary of Theorem 6.

Corollary 9. *There is no oblivious γ-stable ρ-approximation algorithm for* Minimum s-t Cut *in planar graphs for $\rho < \lfloor \frac{n-1}{3} \rfloor$ and $\gamma < \lfloor \frac{n-1}{3} \rfloor$.*

3.3 Amortized Analysis

3.3.1 Approximate Global Cuts

Using a similar construction as in the proof of Theorem 3 it is not hard to obtain an $\Omega(\log n / \log \rho)$ lower bound on the amortized stability of maintaining a ρ-approximate cut in a graph. In the following, however, we derive an even better bound.

Theorem 7. *Any dynamic ρ-approximation algorithm for* Minimum Cut *in general graphs has average stability $\Omega(n/\rho^2)$, even in the clairvoyant setting.*

Proof: We present a sequence of graph instances $(G_1, \ldots, G_n)$ for which maintaining a ρ-approximate cut requires $\Omega(n/\rho^2)$ vertex flips on average. We assume that $\rho = o(\sqrt{n})$ since, otherwise, the theorem is trivial. In the following, we use $V(t)$ to denote the vertex set of graph instance G_t.

Consider the vertex arrival sequence $\sigma = (v_1, \ldots, v_n)$ where, at time $t = 2n/3$, the graph G_t consists of two cliques A and B of equal size $n/3$, with $\rho + 1$ edges between them. Notice that at this time, the cut $X(t) = (A, B)$ is a minimum cut, and is in fact the only ρ-approximate cut available. We partition the rest of the sequence $(v_{t+1}, \ldots, v_n)$ into $\frac{n}{3(1+(\rho+1)^2)}$ batches b_i of size $\ell = 1 + (\rho + 1)^2$. We will argue that for each batch, there is a sequence of vertex arrivals such that any algorithm must perform $\Omega(n)$ vertex flips.

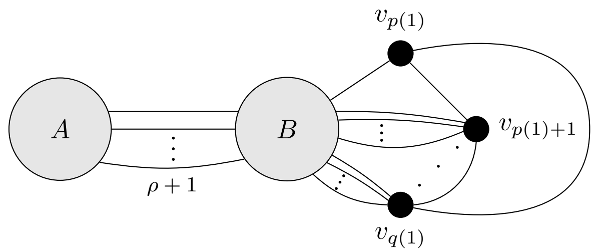

Figure 7: Graph G_t for the proof of the lower bound in Theorem 7 at time $t = 2n/3 + \ell + 1$; that is, after the arrival of the vertices in the first batch $b_1 = (v_{p(1)}, \ldots, v_{q(1)})$. Cliques A and B are highlighted in gray, while vertices in b_1 are highlighted in black.

Let $b_i = (v_{p(i)}, \ldots, v_{q(i)})$ denote the vertex arrival sequence of the i-th batch, with $p(i) = (t+1) + \ell \cdot (i-1)$ and $q(i) = p(i) + \ell$. The vertices arrive as follows. First, vertex $v_{p(i)}$ arrives with an edge to an arbitrary vertex in $V(p(i)-1) \setminus A$ and no other incident vertices. At this point in time, the cut $X(p(i)) = (\{v_{p(i)}\}, V(p(i)-1))$ is a minimum cut, and the only available ρ-approximate cut. Now, each new vertex arriving at time $j \in [p(i)+1, q(i)]$ has edges to all other vertices in $V(j-1) \setminus A$. Notice that at the end of the batch—that is, at time $q(i)$—the cut $X(q(i)) = (A, V(q(i)) \setminus A)$ will be the only available ρ-approximate cut since any other cut must partition either the set $V(q(i)) \setminus A$, the set A, or both and thus has value at least $(\rho+1)^2$. See Figure 7 for an illustration of the graph after the arrival of vertex $v_{q(1)}$.

The claim that $\Omega(n)$-many vertex flips are required in a batch b_i follows from the fact that any algorithm must maintain cuts $X(p(i))$ and $X(q(i))$ at times $p(i)$ and $q(i)$, respectively. Then, by definition of difference between two cuts, we have that: (i) for any batch b_i we have $D(X(p(i)), X(q(i))) = |A|$, and (ii) for any two consecutive batches b_i and b_{i+1} we have $D(X(q(i)), X(p(i+1))) = |A|$. (Notice that, at the start of the first batch b_1, we also have $D(X(t), X(p(1))) = |A|$, with $t = 2n/3$). In other words, in every batch, all the vertices in the set A must be flipped twice.

Now, since there are $\frac{n}{3(1+(\rho+1)^2)}$ batches and each one performs $\Omega(n)$-many vertex flips, the theorem follows. □

3.3.2 Approximate s-t Cuts

A similar construction to that in the proofs of Theorem 3 and Theorem 7 can be used to establish an $\Omega(\log n / \log \rho)$ lower bound on the average number of vertex flips required to maintain a ρ-approximate s-t cut. Unlike the case for Minimum Cut, we show later in Section 3.4 that this bound is tight for Minimum s-t Cut.

Theorem 8. *Any dynamic ρ-approximation algorithm for* Minimum s-t Cut *in general graphs has average stability $\Omega(\log n / \log \rho)$, even in the clairvoyant setting.*

Proof: The proof follows a structure similar to that of Theorem 3. First, construct a graph G_n consisting of vertices s and t, along with three cliques A, B, and C of equal size $\Theta(n)$ connected as described in the proof of Theorem 7. Additionally, include a set D of $\Theta(n)$ vertices, each connected to every vertex in clique B, similar to Figure 3 (with B and C interchanged). At this point, we have not yet defined the edges between the vertices in D and those in the cliques A and C. These are specified next.

Consider the vertex arrival sequence $\sigma = (v_1, v_2, \ldots, v_n)$ where the vertices in $A \cup B \cup C$ all arrive in the prefix subsequence $\sigma_1 = (v_1, v_2, \ldots, v_\ell)$, and vertices in D arrive according to subsequence $\sigma_2 = (v_{\ell+1}, \ldots, v_n)$. The edges between the vertices in D and cliques A and C are now defined as follows: for each vertex $v_{\ell+i} \in D$, connect $v_{\ell+i}$ to $O(\rho^i)$ arbitrary vertices in A if i is odd and to $O(\rho^i)$ arbitrary vertices in C if i is even. (These connections replace the setup in Theorem 3, where vertices in D connected to a fixed number of vertices in A or C.) This construction ensures that the minimum s-t cut value grows geometrically with i.

As in Theorem 3, each vertex arrival in σ_2 induces $\Omega(n)$ vertex flips as the unique ρ-approximate s-t cut oscillates between $(A, \overline{A})$ and $(C, \overline{C})$ until one of these cuts reaches a size of order $O(n)$. The number of iterations exhibiting this oscillating behavior (the size of σ_2) is $\Theta(\log n / \log \rho)$. Thus, asymptotically, this provides a lower bound on the amortized stability of maintaining a ρ-approximate cut in a graph. □

3.4 Algorithms with Low Stability

3.4.1 Approximate Global Cuts

Theorem 4 is tight with respect to the trivial upper bound of $\frac{n-1}{2}$ vertex flips. However, as Theorem 9 below shows, when the approximation factor of the maintained cut is large, very simple and stable algorithms exist. First, we prove the following lemma. We say that a cut $(X, \overline{X})$ is a *singleton* cut if one of X or $\overline{X}$ consists of a single vertex.

Lemma 1. *Any graph $G = (V, E)$ has an $\frac{n-1}{2}$-approximate cut that is a singleton.*

Proof: Consider the singleton cut induced by the vertex of minimum degree, and let $d_{\min}$ be its degree. Now consider an optimal cut (A, B). Define $m := |A|$ and assume without loss of generality that $m \le n/2$. Since $d_{\min} \le n - 1$, we are done when the value of a minimum cut is at least 2, so assume that it is 1. Note that A has at most $\binom{m}{2}$ internal edges, accounting for a total degree of $2 \cdot \binom{m}{2}$. Since each vertex in A has degree at least $d_{\min}$, we thus have at least $m \cdot d_{\min} - m(m-1)$ edges crossing the cut. As we just assumed that the minimum cut value is 1, it follows that $m \cdot d_{\min} - m(m-1) \le 1$, which implies that $d_{\min} \le (m-1) - \frac{1}{m} < \frac{n-1}{2}$. □

Lemma 1 immediately implies that there is a 2-stable $\frac{n-1}{2}$-approximation algorithm, namely the algorithm that maintains a singleton cut that gives a $\frac{n-1}{2}$-approximation. (Note that switching between two singleton cuts requires at most two vertex flips.) Thus we obtain the following theorem.

Theorem 9. *There is a 2-stable $\frac{n-1}{2}$-approximation algorithm for* Minimum Cut *in general graphs.*

For planar graphs, a similar situation occurs when $\rho \ge 5$, since we know that any planar graph has a vertex of degree at most 5. A singleton cut consisting of such a vertex is thus a 5-approximate cut, and maintaining one such cut requires at most two vertex flips per iteration.

Theorem 10. *There is a 2-stable 5-approximation algorithm for* Minimum Cut *in planar graphs.*

Contrary to the case of maintaining a minimum cut, the $\Omega(n/\rho^2)$ amortized lower bound of Theorem 7 is not tight with respect to the trivial $O(n)$ upper bound. We now reduce this gap by showing a new upper bound for maintaining a ρ-approximate cut in the clairvoyant setting.

Theorem 11. *There exists a clairvoyant ρ-approximation algorithm for* Minimum Cut *with amortized stability $O\left(\frac{n \log n}{\rho \log \rho}\right)$.*

For the sake of clarity, we introduce a slightly different notation. We use $v(t)$ to denote the vertex arriving at time t and let $G(t) := (V(t), E(t))$ represent the graph obtained after the arrival of vertex $v(t)$. We let $\mathrm{OPT}(t)$ denote the value of a minimum cut in $G(t)$, and let $\mathrm{ALG}(t)$ denote the value of the cut maintained by our algorithm at time t. To identify a cut in $G(t)$, we specify only the bipartition set $X(t) \subset V(t)$ that contains vertex $v(1)$, and use $\mathrm{cost}(X(t))$ to denote its value in $G(t)$. We use $D(X(t), X(t+1))$ to denote the difference—as defined in the introduction—between cuts $X(t)$ and $X(t+1)$.

Next, we state two simple results that form the basis of our algorithm.

Lemma 2. *Let $X(t)$ be any cut in $G(t)$ and let $t' \leq t$. If $V(t') \not\subseteq X(t)$ then the set $Y(t') = V(t') \cap X(t)$ is a feasible cut in $G(t')$ with $\mathrm{cost}(Y(t')) \leq \mathrm{cost}(X(t))$.*

Proof: Note that $V(t') \cap X(t) \neq \emptyset$ since $v(1) \in V(t') \cap X(t)$. Moreover, $V(t') \cap \overline{X}(t) \neq \emptyset$ since $V(t') \not\subseteq X(t)$. Hence, $Y(t')$ is a feasible cut. Further, the edges crossing the cut $Y(t')$ must be a subset of the edges crossing the cut $X(t)$, hence the value of the cut $Y(t')$ cannot be greater than that of $X(t)$. □

Lemma 3. *If $\mathrm{OPT}(t+1) < \mathrm{OPT}(t)$, then the cut $X(t+1) = V(t)$ is the unique minimum cut in $G(t+1)$.*

Proof: For the sake of contradiction, suppose there is a cut $Y(t+1) \subset V(t)$ with $\mathrm{cost}(Y(t+1)) \leq \mathrm{OPT}(t+1)$. But then the cut $Y(t) = Y(t+1) \setminus \{v(t+1)\}$ has $\mathrm{cost}(Y(t)) < \mathrm{OPT}(t+1) \leq \mathrm{OPT}(t)$, contradicting that $\mathrm{OPT}(t)$ is minimum. □

The algorithm. Consider the sequence $OPT(2), \ldots, OPT(n)$ of minimum cut values at times $t = 2, \ldots, n$. We can partition the time interval $[2, n]$ into sub-intervals, or *phases*, I_i such that $OPT(t)$ is non-decreasing for all $t \in I_i$. Notice that, by Lemma 3, the minimum cut at the start of a phase is always a singleton cut. Now, let s be a parameter. With the aid of clairvoyance, the algorithm can distinguish between two types of phases: a *short phase*—when $|I_i| \leq s$—and a *long phase*—when $|I_i| > s$.

In a short phase $I_{\mathrm{short}} = [t_{\mathrm{start}}, t_{\mathrm{end}}]$, the algorithm adopts the following simple strategy: For all $t \in I_{\mathrm{short}}$, maintain the cut $X_{\mathrm{alg}}(t) = V(t) \setminus \{v(t_{\mathrm{start}})\}$.

Lemma 4. *For any short phase $I_{\mathrm{short}} = [t_{\mathrm{start}}, t_{\mathrm{end}}]$, we have:*

1. *$D(X_{\mathrm{alg}}(t), X_{\mathrm{alg}}(t+1)) = 0$ for all $t \in [t_{\mathrm{start}}, t_{\mathrm{end}} - 1]$, and*
2. *$\mathrm{ALG}(t) \leq (s-1) \cdot \mathrm{OPT}(t)$ for all $t \in I_{\mathrm{short}}$.*

Proof: The first part of the lemma trivially follows from the definition of cut difference and the fact that $X_{\mathrm{alg}}(t+1) \cap X_{\mathrm{alg}}(t) = \{v(t)\}$. The second part follows from the fact that by Lemma 3, the starting cut $X_{\mathrm{alg}}(t_{\mathrm{start}})$ is minimum, and from observing that each vertex arrival after t_{start} can only increase the degree of $v(t_{\mathrm{start}})$ in one unit. Putting this together with the fact that $\mathrm{OPT}(t) \geq \mathrm{OPT}(t_{\mathrm{start}})$ for all $t \in I_{\mathrm{short}}$ implies the result. □

Now, let I_{long} be a long phase, and let ρ be the approximation guarantee we want to achieve with the algorithm. We define a *sub-phase* $I_{\mathrm{sub}} = [t_{\mathrm{start}}, t_{\mathrm{end}}]$ of phase I_{long} as a maximal time interval such that $\mathrm{OPT}(t_{\mathrm{end}}) \leq \rho \cdot \mathrm{OPT}(t_{\mathrm{start}})$, with $t_{\mathrm{start}}, t_{\mathrm{end}} \in I_{\mathrm{long}}$. See Figure 8 for an illustration. (With a slight abuse of notation, we are re-using the notation t_{start} and t_{end} here, to also denote the start and end of a sub-phase.) Notice that there can be up to $O(\log n / \log \rho)$ sub-phases in a long

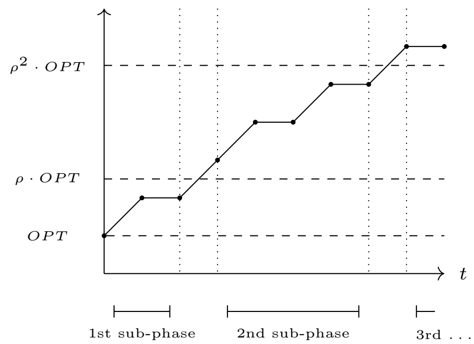

Figure 8: Illustration of a long phase and its division into sub-phases based on the approximation guarantee, ρ. The y-axis represents the range $[1, n]$ of possible min-cut values, with OPT denoting the min-cut value at the start of the long phase. The x-axis represents discrete time steps, partitioned by dotted vertical lines that divide the long phase into sub-phases.

phase. Ideally, we would like our algorithm to identify sub-phases in a long phase and for each sub-phase adopt the following strategy: For all $t \in I_{\text{sub}}$, maintain the cut $X_{\text{alg}}(t) = V(t) \cap X_{\text{opt}}(t_{\text{end}})$, where $X_{\text{opt}}(t_{\text{end}})$ is a minimum cut in $G(t_{\text{end}})$. This has the potential to grant us similar results to Lemma 4. The strategy, however, is flawed: the cut $X_{\text{alg}}(t)$ might be infeasible since there can be some time $t' \in I_{\text{sub}}$ for which $V(t') \subseteq X_{\text{opt}}(t_{\text{end}})$.

To refine this strategy, we further partition a sub-phase $I_{\text{sub}} = [t_{\text{start}}, t_{\text{end}}]$ into sub-intervals $I^i_{\text{sub}} = [t_i, t_{i+1})$ as follows. First, we let $t_0 := t_{\text{start}}$. Then, given t_i, we define t_{i+1} as the time immediately after the "furthest" time $t \in I_{\text{sub}}$ such that $X_{\text{opt}}(t)$—a minimum cut in $G(t)$—induces a feasible cut in $G(t_i)$. More formally, $t_{i+1} = 1 + \max\{t \mid t \leq t_{\text{end}} \text{ and } V(t_i) \nsubseteq X_{\text{opt}}(t)\}$. Now, as our new strategy, for each sub-interval $I^i_{\text{sub}} = [t_i, t_{i+1})$ let the algorithm perform the following: For all $t \in I^i_{\text{sub}}$, maintain the cut $X_{\text{alg}}(t) = V(t) \cap X_{\text{opt}}(t_{i+1} - 1)$.

Lemma 5. *For any sub-phase $I_{\text{sub}} = [t_{\text{start}}, t_{\text{end}}]$ of a long phase we have:*

1. $\sum_t D(X_{\text{alg}}(t), X_{\text{alg}}(t+1)) = O(n)$ *for all* $t \in [t_{\text{start}}, t_{\text{end}} - 1]$, *and*

2. $\text{ALG}(t) \leq \rho \cdot \text{OPT}(t)$ *for all* $t \in I_{\text{sub}}$.

Proof: We start with the second part of the lemma. First, observe that $X_{\text{alg}}(t)$ is feasible throughout I_{sub}, since on each sub-interval $[t_i, t_{i+1})$ of I_{sub} we have that $X_{\text{opt}}(t_{i+1} - 1)$ induces a feasible cut on $G(t_i)$, hence also on $G(t)$ for all $t \in [t_i, t_{i+1})$. Now, for each sub-interval I^i_{sub} of I_{sub}, by Lemma 2 we have $\text{ALG}(t) \leq \text{OPT}(t_{i+1} - 1)$ for all $t \in I^i_{\text{sub}}$. But $\text{OPT}(t_{i+1} - 1) \leq \rho \cdot \text{OPT}(t_{\text{start}})$ for every sub-interval of I_{sub}. Hence, $\text{ALG}(t) \leq \rho \cdot \text{OPT}(t_{\text{start}})$ for all $t \in I_{\text{sub}}$.

Now we prove the first part of the lemma. Recall that $v(1) \in X_{\text{alg}}(t)$ for all $t \in I_{\text{sub}}$. First, observe that any vertex is placed into $X_{\text{alg}}(t)$ at most once during any given sub-interval. (It is simply assigned to the maintained set $X_{\text{alg}}(t)$ of the bipartition or its complement.) Now, let $I^i_{\text{sub}} = [t_i, t_{i+1})$ be a sub-interval of I_{sub}. We claim that a vertex $v \in X_{\text{alg}}(t_i)$ cannot be flipped out of $X_{\text{alg}}(t)$ in any t such that $t_{i+1} \leq t \leq t_{\text{end}}$. This follows because, otherwise, there would be a time $t' \geq t_{i+1} - 1$ such that $X_{\text{opt}}(t')$ induces a feasible cut in $G(t_i)$, which violates the condition that $t_{i+1} - 1$ was maximal. Therefore, a vertex can be flipped in the sub-phase I_{sub} at most once. Accounting for all vertices then gives the result. □

We are now ready to prove Theorem 11.

Proof: The approximation ratios of short and long phases are $(s-1)$ and ρ, respectively. Hence, the approximation ratio of the algorithm is $\max(s-1,\rho)$. Now we analyze the stability of the algorithm. First observe that, by Lemma 4, there are no vertex flips performed in short phases. As for long phases, we know that each can have at most $O(\frac{\log n}{\log \rho})$ sub-phases and, by Lemma 5, each sub-phase performs at most $O(n)$ vertex flips in total. There are at most $\frac{n}{s}$ long phases, hence the total number of vertex flips performed by long phases is $\frac{n}{s} \cdot O(n \cdot \frac{\log n}{\log \rho}) = O(\frac{n^2 \log n}{s \log \rho})$.

We only have left to account for the number of vertex flips induced at the start of each phase and sub-phase; namely, when going from one phase (resp. sub-phase) to the next. Notice that, by Lemma 3, going from a short phase to another phase (either short or long) induces a single vertex flip. On the other hand, going from a long phase to a short phase, as well as from a long phase to another long phase, can each induce $O(n)$ vertex flips. Hence, the total number of vertex flips performed at the start of long phases is $O(\frac{n^2}{s})$. Finally, within each long phase, the total number of vertex flips performed when going from the end of one subphase to the beginning of the next subphase is $O(n \cdot \frac{\log n}{\log \rho})$. In total for every long phase then, we have $O(\frac{n^2 \log n}{s \log \rho})$ vertex flips.

Putting all this together, we get that the total number of vertex flips performed by the algorithm is $O(\frac{n^2 \log n}{s \log \rho})$. The result of the theorem then follows by setting $s = \rho$ and dividing the total number of vertex flips by n. $\square$

3.4.2 Approximate s-t Cuts

One of the main challenges in designing stable algorithms for Minimum Cut is that a singleton cut, consisting of a newly revealed vertex, can become the minimum cut. (Lemma 3 shows this is the only way the minimum cut value can decrease in the vertex arrival model.) For Minimum s-t Cut, however, this issue does not arise, since the only singleton s-t cuts are those defined by s or t. Consequently, the value of the minimum s-t cut increases monotonically during a vertex arrival sequence. Using this property, we now prove that the clairvoyant algorithm of Theorem 11 can be used to establish a tight bound on the amortized stability of maintaining a ρ-approximate s-t cut. We use the notation established for proving Theorem 11, most notably, the notions of long phases and sub-phases.

Theorem 12. *There exists a clairvoyant ρ-approximation algorithm for* Minimum s-t Cut *with amortized stability* $O\left(\frac{\log n}{\log \rho}\right)$.

Proof: Consider the sequence $OPT(i_{\text{path}}), \ldots, OPT(n)$ of minimum s-t cut values at times $i \in [p, n]$, where i_{path} denotes the first timestep at which s and t are joined by a path. Since the minimum s-t cut value is monotonically increasing during a vertex arrival sequence, we treat the time interval $I_{\text{long}} = [i_{\text{path}}, n]$ in the same way as a long phase in the clairvoyant algorithm of Theorem 11. Hence, we partition I_{long} into $O(\log n / \log \rho)$ sub-phases. Recall that a sub-phase $I_{\text{sub}} = [i_{\text{start}}, i_{\text{end}}]$ of I is a maximal time interval such that $OPT(i_{\text{end}}) \leq \rho \cdot OPT(i_{\text{start}})$.

With clairvoyance, our algorithm identifies all sub-phases of I. For each sub-phase $I_{\text{sub}} = [i_{\text{start}}, i_{\text{end}}]$, the algorithm adopts the following strategy: For all $i \in [i_{\text{start}}, i_{\text{end}}]$, maintain the cut $X_{\text{alg}}(i) = V(i) \cap X_{\text{opt}}(i_{\text{end}})$, where $X_{\text{opt}}(i_{\text{end}})$ is a minimum s-t cut in the graph $G(i_{\text{end}})$. That is, the algorithm looks ahead at timestep i_{end}, finds a minimum s-t cut $X_{\text{alg}}(i_{\text{end}})$, and keeps the same bipartition throughout $i \in [i_{\text{start}}, i_{\text{end}}]$ limited to the vertices revealed until timestep i. Note that the maintained s-t cut is always feasible, as no side of the bipartition is ever empty. Moreover, from Lemma 5, we have that $ALG(i) \leq \rho \cdot OPT(i)$, for all $i \in I_{\text{sub}}$.

In terms of stability, the difference between the s-t cuts maintained at consecutive timesteps within a sub-phase is 0, meaning no vertex flips are performed during the sub-phases. However, vertex flips may occur when transitioning between sub-phases. As shown in the proof of Theorem 11, the total number of vertex flips caused by transitions between sub-phases in a long phase is $O\left(\frac{n \log n}{\log \rho}\right)$. Dividing this total by n gives the desired result. $\square$

4 Concluding Remarks

We studied the stability of dynamic algorithms for Minimum Cut and Minimum s-t Cut under the vertex-arrival model. We showed that, for general and planar graphs, the trivial stability bound is tight up to constant factors in both the oblivious and clairvoyant settings. This holds for maintaining both exact and ρ-approximate cuts. When the approximation ratio satisfies $\rho \geq \frac{n-1}{2}$ in general graphs and $\rho \geq 5$ in planar graphs, we show that there are simple 2-stable ρ-approximation algorithms for Minimum Cut. In the amortized case, we also obtained that the trivial stability bound is tight up to constant factors for both problems, but only for the exact case. When maintaining ρ-approximate cuts, we showed that there are better-than-trivial average stability bounds. For approximating Minimum Cut, we establish a lower bound of $\Omega(n/\rho^2)$ and give a clairvoyant algorithm with amortized stability $O\left(\frac{n \log n}{\rho \log \rho}\right)$. For approximating Minimum s-t Cut, we establish a tight stability bound of $\Theta\left(\frac{\log n}{\log \rho}\right)$ in the clairvoyant setting.

The lower bound proofs in this work rely on specific constructions that may never show up in practice. We believe that situations in which a vertex insertion induces many vertex flips are rare. As such, the average case analysis of amortized stability of global and s-t min-cuts seems like an interesting research direction. This is further motivated by the average-case results obtained in this work. Another promising approach toward improved stable approximation algorithms for global and s-t min-cuts is to consider graphs of bounded degree. Finally, we believe that exploring other problems from the viewpoint of stability is an interesting endeavor.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] A. Bernstein and A. Dudeja. Online matching with recourse: random edge arrivals. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, pages 11:1–11:16, 2020. doi:10.4230/LIPIcs.FSTTCS.2020.11.

[3] M. de Berg, A. López Martínez, and F. Spieksma. Stable and dynamic minimum cuts. In R. Uehara, K. Yamanaka, and H.-C. Yen, editors, *WALCOM: Algorithms and Computation*, pages 273–287, Singapore, 2024. Springer Nature Singapore.

[4] M. de Berg, A. Sadhukhan, and F. Spieksma. Stable approximation algorithms for dominating set and independent set. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, pages 27:1–27:19, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.27.

[5] M. de Berg, A. Sadhukhan, and F. Spieksma. Stable approximation algorithms for the dynamic broadcast range-assignment problem. *SIAM Journal on Discrete Mathematics*, 38(1):790–827, 2024. doi:10.1137/23M1545975.

[6] B. Feldkord, M. Feldotto, A. Gupta, G. Guruganesh, A. Kumar, S. Riechers, and D. Wajc. Fully-dynamic bin packing with little repacking. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, pages 51:1–51:24, 2018. doi:10.4230/LIPIcs.ICALP.2018.51.

[7] G. Goranci, M. Henzinger, D. Nanongkai, T. Saranurak, M. Thorup, and C. Wulff-Nilsen. Fully dynamic exact edge connectivity in sublinear time. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 70–86. SIAM, 2023. doi:10.1137/1.9781611977554.ch3.

[8] G. Goranci, M. Henzinger, and M. Thorup. Incremental exact min-cut in polylogarithmic amortized update time. *ACM Transactions on Algorithms (TALG)*, 14(2):1–21, 2018. doi:10.1145/3174803.

[9] A. Gu, A. Gupta, and A. Kumar. The power of deferral: maintaining a constant-competitive steiner tree online. In *Proceedings of the 45th annual ACM symposium on Theory of Computing*, pages 525–534, 2013. doi:10.1145/2488608.2488674.

[10] A. Gupta, A. Kumar, and C. Stein. Maintaining assignments online: Matching, scheduling, and flows. In *Proceedings of the 25th annual ACM-SIAM symposium on Discrete algorithms*, pages 468–479. SIAM, 2014. doi:10.1137/1.9781611973402.35.

[11] X. Han and K. Makino. Online minimization knapsack problem. *Theoretical Computer Science*, 609:185–196, 2016. doi:10.1016/j.tcs.2015.09.021.

[12] M. Hasheminezhad, B. D. McKay, and T. Reeves. Recursive generation of simple planar 5-regular graphs and pentangulations. *Journal of Graph Algorithms and Applications*, 15(3):417–436, 2011. doi:10.7155/jgaa.00232.

[13] M. Henzinger, A. Noe, and C. Schulz. Practical fully dynamic minimum cut algorithms. In *Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 13–26, 2022. doi:10.1137/1.9781611977042.2.

[14] M. Imase and B. M. Waxman. Dynamic steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3):369–384, 1991. doi:10.1137/0404033.

[15] N. Megow, M. Skutella, J. Verschae, and A. Wiese. The power of recourse for online mst and tsp. *SIAM Journal on Computing*, 45(3):859–880, 2016. doi:10.1137/130917703.

[16] M. Thorup. Fully-dynamic min-cut. *Combinatorica*, 27(1):91–127, 2007. doi:10.1007/s00493-007-0045-2.

[17] M. Thorup and D. R. Karger. Dynamic graph algorithms with applications. In *Scandinavian Workshop on Algorithm Theory*, pages 1–9, 2000. doi:10.1007/3-540-44985-X_1.

[18] O. Wasim and V. King. Fully dynamic sequential and distributed algorithms for max-cut. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, pages 33:1–33:19, 2020. doi:10.4230/LIPIcs.FSTTCS.2020.33.