Stable and Dynamic Minimum Cuts

Stable and Dynamic Minimum Cuts

Mark de Berg, Andrés López Martínez(✉), and Frits Spieksma

Department of Mathematics and Computer Science, TU Eindhoven, Eindhoven, The Netherlands
{m.t.d.berg,a.lopez.martinez,f.c.r.spieksma}@tue.nl

Abstract. We consider the problems of maintaining exact *minimum cuts* and *ρ-approximate cuts* 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 the two problems, in general graphs as well as in planar graphs, 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 relaxed more than certain thresholds, 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 exact 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 $O\left(\frac{n \log n}{\rho \log \rho}\right)$.

Keywords: Dynamic Minimum Cut · Stability · Approximation

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 $\overline{S}$. The *size* or *value* of the cut, denoted by $w(S, \overline{S})$, is the total number of edges connecting a node in S with a node in $\overline{S}$. A *minimum cut*, or *min-cut*, is a cut with the smallest size. Finding such a cut is a classic combinatorial optimization problem and has numerous practical and theoretical applications [1]. Throughout, we call this problem Minimum Cut. For a positive integer k, graph G is said to be *k-edge connected* if every cut in G has size at least k. Let $\lambda(G)$ denote the value of a min-cut in G. For a parameter $\rho \geq 1$, a ρ-approximate cut $(X, \overline{X})$ is a cut with value at most $\rho\lambda(G)$; that is, $w(X, \overline{X}) \leq \rho\lambda(G)$.

This research was 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.

R. Uehara et al. (Eds.): WALCOM 2024, LNCS 14549, pp. 273–287, 2024.
https://doi.org/10.1007/978-981-97-0566-5_20

We consider the problem of maintaining an exact or approximate min-cut in the *vertex-arrival model*, where the graph G is subject to changes over time due to new vertices being inserted into G. Starting with the empty graph G_0, new vertices arrive *one by one* together with *all* their incident edges to previously arrived vertices, thus producing a sequence of graph instances $(G_0, G_1, \ldots, G_n)$ with n the number of vertices in G_n. We further assume that each graph in the sequence is connected. 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.*[2], 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 solutions in dynamic graph cuts 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 $D(S_i, S_{i+1}) = \min(|X \triangle Y|, |X \triangle \overline{Y}|)$.) 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. [9] and Bernstein et al. [4] for work on maintaining matchings and flows, Imase and Waxman [13], Megow *et al.*[14] and Gu *et al.*[8] for work on maintaining (Steiner) trees and Hamiltonian cycles, and Feldkord *et al.*[5] and Han and Makino [10] 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 [6,7,12,15,16]. There is also work that focuses on the "difference" between two consecutive solutions, while not taking explicitly computational time into account; we mention Wasim and King [17] for work on Max-Cut, and De Berg et al. [3] 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 in the vertex-arrival model. More precisely, we obtain lower and upper bounds on the stability of dynamic min-cuts on general graphs as well as planar graphs. The results are summarized in Table 1.

Table 1. Summary of results on γ-stability for Minimum Cut.

Graph class	Exact		ρ-Approximation	
	Lower bound	Upper bound	Lower bound	Upper bound
General	$\frac{n-1}{2}$	$\frac{n-1}{2}$	$\frac{n-2}{2}$ (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 $\rho > \frac{n-1}{2}$)
Planar	$\frac{n-1}{2}$	$\frac{n-1}{2}$	$\frac{n-2}{2}$ (for $\rho < 5$)	$\frac{n-1}{2}$ (for $\rho < 5$)
			0 (for $\rho \geq 5$)	2 (for $\rho \geq 5$)
General (amortized)	$\frac{n}{16}$	$\frac{n-1}{4}$	$\Omega(n/\rho^2)$	$O\left(\frac{n \log n}{\rho \log \rho}\right)$

For general graphs, we show that an algorithm maintaining an exact minimum cut may need $\frac{n-1}{2}$ vertex flips in each iteration. This result is tight (as one can always change from one cut to another one using at most $\frac{n-1}{2}$ vertex flips), and applies to both the *oblivious setting*—when the algorithm has no knowledge of the vertex-arrival sequence other than the previously arrived elements—and the *clairvoyant setting*—when the algorithm is allowed to see the entire sequence of vertex arrivals in advance. Similar results apply to the special case of planar graphs. In contrast, the problem becomes 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 an exact minimum cut, $\Theta(n)$ vertex flips are needed.

We now turn to the case of maintaining a ρ-approximate cut. For general graphs, we show that similar to the exact case, an algorithm may need $\frac{n-2}{2}$ vertex flips in each 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 $O\left(\frac{n \log n}{\rho \log \rho}\right)$.

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

2 Maintaining Exact 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 $n \geq 9$ such that $\gamma < \left\lfloor \frac{n-1}{2} \right\rfloor$.*

Proof. For every $n \geq 9$, we present a sequence of graph instances $(G_1, \ldots, G_n)$ (see Fig. 1) for which an exact stable algorithm requires at least $\left\lfloor \frac{n-1}{2} \right\rfloor$ vertex flips to maintain an exact 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. □

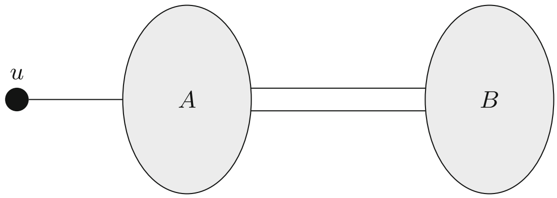

Fig. 1. Graph G_n for the proof of the lower bound in Theorem 1. Cliques A and B (gray) are connected by means of two edges between arbitrary vertices. Vertex u (black) is connected to an arbitrary vertex in clique A. After its arrival, any algorithm must perform $\frac{n-1}{2}$ vertex flips to maintain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

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. 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 $n \geq 9$ such that $\gamma < \left\lfloor \frac{n-1}{2} \right\rfloor$.*

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 such that $\gamma < \left\lfloor \frac{n-1}{2} \right\rfloor$, 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). ☐

Amortized Analysis. 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 stability 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 2. *There is no stable and 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 $(G_1, \ldots, G_n)$ for which an exact stable algorithm requires at least $\frac{n}{16} - 2$ vertex flips *on average* (hence $\Omega(n^2)$ flips *in total*) to maintain an exact 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 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 Fig. 2 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.

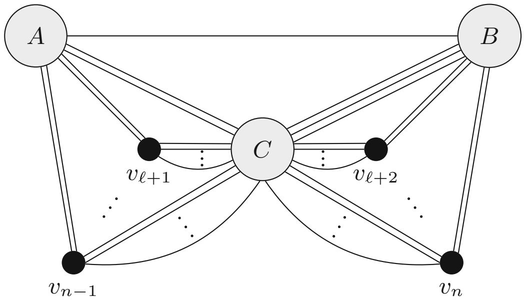

Fig. 2. Graph G_n for the proof of the lower bound in Theorem 2. Cliques A, B, and C are highlighted in gray. The vertices in set D are highlighted in black. Of these vertices, those to the left (resp. right) of clique C belong to the set D_1 (resp. D_2). The alternated arrival of vertices from D_1 and D_2 induce $\Omega(n)$ vertex flips per iteration.

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 2 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 $\frac{n-1}{4}$.

3 Maintaining Approximate 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

[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 $n > 5$.

surprisingly, the following result shows that no better trade-off can be achieved for approximate solutions.

Theorem 3. *There is no γ-stable ρ-approximation algorithm for* Minimum Cut *in general graphs of size $n \geq 10$ such that $\rho < \lfloor \frac{n-2}{2} \rfloor - 2$ and $\gamma < \lfloor \frac{n-2}{2} \rfloor$.*

Proof. For every $n \geq 10$, we present a sequence of graph instances $(G_1, \ldots, G_n)$ for which a stable approximation algorithm requires at least $\lfloor \frac{n-2}{2} \rfloor$ vertex flips to obtain an approximation ratio less than ℓ where $1 < \ell \leq \lfloor \frac{n-2}{2} \rfloor - 2$ (when $\ell = \lfloor \frac{n-2}{2} \rfloor - 2$ the main claim follows)[2]. The graph G_n has two cliques A and B of roughly equal size such that $|A| \geq \lfloor \frac{n-2}{2} \rfloor$ and $|B| \geq \lfloor \frac{n-2}{2} \rfloor$, 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 Fig. 3 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 $\lfloor \frac{n-2}{2} \rfloor - 1 > \ell$. 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 $\lfloor \frac{n-2}{2} \rfloor$-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 $\lfloor \frac{n-2}{2} \rfloor - 1$ 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 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.)

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

[3] Because any clique-crossing cut has value at least $\lfloor \frac{n-2}{2} \rfloor - 1 > \ell$ since $deg(v) \geq \lfloor \frac{n-2}{2} \rfloor - 1$ $\forall v \in A \cup B$. 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.

As mentioned above, the only valid solution at timestep n is $S_n = (\{u\}, A \cup B \cup \{w\})$, but $D(S_{n-1}, S_n) \geq \left\lfloor \frac{n-2}{2} \right\rfloor$; that is, cut S_n is at least $\left\lfloor \frac{n-2}{2} \right\rfloor$ vertex flips away from S_{n-1}. Therefore, at least $\left\lfloor \frac{n-2}{2} \right\rfloor$ 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 $D(S_{n-2}, S_{n-1}) \geq \left\lfloor \frac{n-2}{2} \right\rfloor$, because $S_{n-2} = (A, B)$. Therefore, at least $\left\lfloor \frac{n-2}{2} \right\rfloor$ vertex flips are performed at timestep $n - 1$.

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

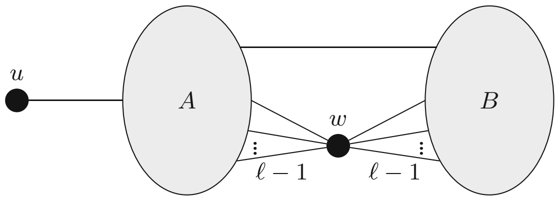

Fig. 3. Graph G_n for the proof of the lower bound in Theorem 3. Cliques A and B (gray) are connected by a single edge between arbitrary vertices. Vertex u (black) is connected to an arbitrary vertex in clique A, and vertex w (black) to $\ell - 1$ arbitrary vertices in A and B, respectively. After the arrival of both u and w, any dynamic algorithm must perform $\frac{n-2}{2}$ vertex flips to maintain a ρ-approximate cut.

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

Corollary 3. *There is no clairvoyant γ-stable ρ-approximation algorithm for* Minimum Cut *in general graphs of size $n \geq 10$ such that $\rho < \left\lfloor \frac{n-2}{2} \right\rfloor - 2$ and $\gamma < \left\lfloor \frac{n-2}{2} \right\rfloor$.*

Proof. This follows directly from the proof of Theorem 3. 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 $\left\lfloor \frac{n-2}{2} \right\rfloor$ vertex flips in some timestep. □

Similarly, we have the following for planar graphs.

Corollary 4. *There is no γ-stable ρ-approximation algorithm for* Minimum Cut *in planar graphs such that $\rho < 5$ 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. [11, Fig. 1]). Then, we can use the proof of Theorem 3 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. □

Amortized Analysis. Using a similar construction as in the proof of Theorem 2 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 4. *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.

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 $\rho + 1$. See Fig. 4 for an illustration of the graph after the arrival of vertex $v_{q(1)}$.

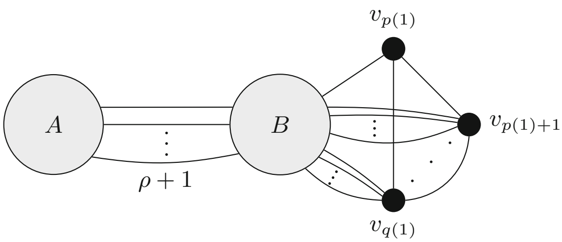

Fig. 4. Graph G_t for the proof of the lower bound in Theorem 4 at time $t = 2n/3+\ell+1$; that is, after the arrival of the vertices in the first batch $b_1 = (v_{p(1)}, \ldots, v_{q(1)})$. Cliques A and B are highlighted in gray, while vertices in b_1 are highlighted in black.

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 main claim follows. □

3.1 Improved Upper Bounds

Theorem 3 is tight with respect to the trivial upper bound of $\frac{n-1}{2}$ vertex flips. However, as Theorem 5 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 $m \leq n/2$. Since $d_{\min} \leq n-1$, 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. But we just observe that the minimum cut value is 1, so $m \cdot d_{\min} - m(m-1) \leq 1$, which implies that $d_{\min} \leq (m-1) - \frac{1}{m} < \frac{n-1}{2}$. □

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 5. *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 $\rho \geq 5$, 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 6. *There is a 2-stable 5-approximation algorithm for* Minimum Cut *in planar graphs.*

Amortized Analysis. Contrary to the case of maintaining an exact minimum cut, the $\Omega(n/\rho^2)$ amortized lower bound of Theorem 4 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 7. *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 $I_{\text{short}} = [t_{\text{start}}, t_{\text{end}}]$, the algorithm adopts the following simple strategy: For all $t \in I_{\text{short}}$, maintain the cut $X_{\text{alg}}(t) = V(t) \setminus \{v(t_{\text{start}})\}$.

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

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

Proof. The first part of the lemma trivially follows from the definition of cut difference and the fact that $X_{\text{alg}}(t+1) \cap X_{\text{alg}}(t) = \{v(t)\}$. The second part follows from the fact that by Lemma 3, the starting cut $X_{\text{alg}}(t_{\text{start}})$ is minimum, and from observing that each vertex arrival after t_{start} can only increase the degree of $v(t_{\text{start}})$ in one unit. Putting this together with the fact that $\text{OPT}(t) \geq \text{OPT}(t_{\text{start}})$ for all $t \in I_{\text{start}}$ 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* $I_{\text{sub}} = [t_{\text{start}}, t_{\text{end}}]$ of phase I_{long} as a maximal time interval such that $\text{OPT}(t_{\text{end}}) \leq \rho \cdot \text{OPT}(t_{\text{start}})$, with $t_{\text{start}}, t_{\text{end}} \in I_{\text{long}}$. (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 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 $I_{\text{sub}}^i = [t_i, t_{i+1})$ 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 $I_{\text{sub}}^i = [t_i, t_{i+1})$ let the algorithm perform the following: For all $t \in I_{\text{sub}}^i$, 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 7.

Proof of Theorem 7. 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. □

4 Concluding Remarks

We studied the stability of dynamic algorithms for Minimum 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, but only for the exact case. When maintaining ρ-approximate cuts, we showed that there are better-than-trivial average stability bounds—namely, a lower bound of $\Omega(n/\rho^2)$ and a clairvoyant algorithm with amortized stability $O\left(\frac{n \log n}{\rho \log \rho}\right)$.

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 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 min-cut 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. Ahuja, R., Magnanti, T., Orlin, J.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, New York (1993)
2. de Berg, M., Sadhukhan, A., Spieksma, F.: Stable approximation algorithms for the dynamic broadcast range-assignment problem. In: 18th Scandinavian Symposium and Workshops on Algorithm Theory, pp. 15:1–15:21 (2022)
3. de Berg, M., Sadhukhan, A., Spieksma, F.: Stable approximation algorithms for dominating set and independent set. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023), pp. 27:1–27:19 (2023)
4. Bernstein, A., Dudeja, A.: Online matching with recourse: random edge arrivals. In: 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020), pp. 11:1–11:16 (2020)
5. Feldkord, B., et al.: Fully-dynamic bin packing with little repacking. In: 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018), pp. 51:1–51:24 (2018)
6. Goranci, G., Henzinger, M., Nanongkai, D., Saranurak, T., Thorup, M., Wulff-Nilsen, C.: Fully dynamic exact edge connectivity in sublinear time. In: Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 70–86. SIAM (2023)
7. Goranci, G., Henzinger, M., Thorup, M.: Incremental exact min-cut in polylogarithmic amortized update time. ACM Trans. Algorithms (TALG) **14**(2), 1–21 (2018)
8. Gu, A., Gupta, A., Kumar, A.: The power of deferral: maintaining a constant-competitive steiner tree online. In: Proceedings of the 45th Annual ACM Symposium on Theory of Computing, pp. 525–534 (2013)
9. Gupta, A., Kumar, A., Stein, C.: Maintaining assignments online: matching, scheduling, and flows. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 468–479. SIAM (2014)

10. Han, X., Makino, K.: Online minimization knapsack problem. In: Bampis, E., Jansen, K. (eds.) Approximation and Online Algorithms. WAOA 2009. LNCS, vol. 5893, pp. 182–193. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12450-1_17
11. Hasheminezhad, M., McKay, B.D., Reeves, T.: Recursive generation of simple planar 5-regular graphs and pentangulations. J. Graph Algorithms Appl. **15**(3), 417–436 (2011)
12. Henzinger, M., Noe, A., Schulz, C.: Practical fully dynamic minimum cut algorithms. In: Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX), pp. 13–26 (2022)
13. Imase, M., Waxman, B.M.: Dynamic steiner tree problem. SIAM J. Discret. Math. **4**(3), 369–384 (1991)
14. Megow, N., Skutella, M., Verschae, J., Wiese, A.: The power of recourse for online MST and TSP. SIAM J. Comput. **45**(3), 859–880 (2016)
15. Thorup, M.: Fully-dynamic min-cut. Combinatorica **27**(1), 91–127 (2007)
16. Thorup, M., Karger, D.R.: Dynamic graph algorithms with applications. In: Scandinavian Workshop on Algorithm Theory, pp. 1–9 (2000)
17. Wasim, O., King, V.: 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), pp. 33:1–33:19 (2020)