Discounted Cuts: A Stackelberg Approach to Network Disruption

Pål Grønås Drange[1], **Fedor Fomin**[1], **Petr A. Golovach**[1], **Danil Sagunov**[2]

[1]University of Bergen, Norway
[2]ITMO University, Russia

Abstract

We study a Stackelberg variant of the classical Most Vital Links problem, modeled as a one-round adversarial game between an attacker and a defender. The attacker strategically removes up to k edges from a flow network to maximally disrupt flow between a source s and a sink t, after which the defender optimally reroutes the remaining flow. To capture this attacker–defender interaction, we introduce a new mathematical model of discounted cuts, in which the cost of a cut is evaluated by excluding its k most expensive edges. This model generalizes the Most Vital Links problem and uncovers novel algorithmic and complexity-theoretic properties.

We develop a unified algorithmic framework for analyzing various forms of discounted cut problems, including minimizing or maximizing the cost of a cut under discount mechanisms that exclude either the k most expensive or the k cheapest edges. While most variants are NP-complete on general graphs, our main result establishes polynomial-time solvability for all discounted cut problems in our framework when the input is restricted to bounded-genus graphs, a relevant class that includes many real-world networks such as transportation and infrastructure networks. With this work, we aim to open collaborative bridges between artificial intelligence, algorithmic game theory, and operations research.

Extended version — https://arxiv.org/abs/2511.10804

1 Introduction

Consider the following scenario of securing a network: An attacker attempts to escalate privileges from server s to a critical database server t. The security team has a limited budget β to deploy firewalls or physically disconnect cables. Some links are very expensive to disable—typically those that are main trunk lines carrying heavy traffic. However, the company's cybersecurity insurance allows the team to discount the cost of disabling up to k links, either by covering the most expensive ones or, alternatively, the least expensive.

As another example, consider a multi-agent system where agents are connected via conflict relationships. We aim to divide the agents into two teams or coalitions in a way that maximizes inter-group tension. When agents are represented as nodes and interaction strengths as weighted edges, this leads to a natural MAXCUT formulation: the goal is to partition the graph to maximize the total weight of edges crossing the cut. However, in many realistic scenarios, some interactions may be negligible or unreliable—such as low-trust links, wrong information, or edges that can be manipulated by a bounded-perception attacker. In such cases, we would be more interested in solving the *discounted maximum cut*, where the objective function ignores the k weakest (i.e., lowest-weight) edges in the cut and thus to focus on strong, strategically meaningful interactions, while discounting minor or noisy ones.

This gives rise to a Stackelberg variant of the classic cut problems, where a leader chooses a cut and a follower modifies the cost of up to k edges post hoc. Depending on whether the follower discounts the most or the least expensive edges, the effective cost of the cut—and thus the optimal strategy—changes significantly. See Figure 1 for an example.

Although the discount variants of cut problems appear closely related to the standard cut problems and to classical interdiction models such as the MOST VITAL LINKS problem (introduced by Wollmer (1963, 1964)), they give rise to novel algorithmic questions. In particular, the cost function becomes non-additive and adversarially dependent on the edge weights, breaking standard structural properties of cuts.

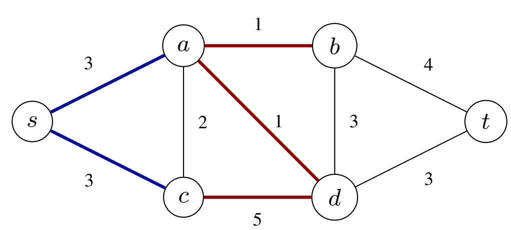

Figure 1: A minimum s-t-cut in this example is $\{sa, sc\}$ (blue) of cost 6. For $k = 1$, the minimum discounted s-t-cut is $\{ab, ad, cd\}$ (red), with discounted cost 2. This corresponds to the edge cd being the *most vital link*, i.e., the removal of cd leads to the largest decrease in maximum flow.

1.1 Cuts with k Free Edges

In this paper, we introduce and study several variants of the minimum and maximum cut problems in which the cost of some edges in the cut is discounted. Specifically, we consider two discount mechanisms: one in which we do not pay for the k most expensive edges, and another in which we do not pay for the k cheapest edges. These correspond to the two natural game strategies of always discounting the most expensive, and most cheap edges. We present these discounted-cost cut problems as formal models of network vulnerability and adversarial flow inhibition, providing a suite of complexity and algorithmic results with a particular focus on tractable cases in graphs of bounded genus. Our findings advance the algorithmic understanding of partial cuts under adversarial or cooperative discounting, opening new directions in network design and interdiction.

Let S be an ordered list of numbers, ordered from smallest to largest. We define two discounted cost functions

$$\mathsf{Cost}_k^{\exp}(S) = \sum_{i}^{k} S_i \quad \text{and} \quad \mathsf{Cost}_k^{\text{cheap}}(S) = \sum_{n-k}^{n} S_i,$$

i.e., $\mathsf{Cost}_k^{\exp}$ (resp. $\mathsf{Cost}_k^{\text{cheap}}$) is the sum of the elements in S *except* the k most (resp. least) expensive elements. In this work we will be talking mostly about cuts, so we define the following. Given a cut (A, B) of a graph G with an (edge) cost function $c\colon E(G) \to \mathbb{Z}_{\geq 0}$ and an integer $k \geq 0$, we define the *discounted cost* of the cut (A, B) as

$$\begin{aligned}\mathsf{Cost}_k^{\exp}(A, B) &= c(E(A, B)) - c(R) \\ &= \sum_{e \in E(A,B)} c(e) - \sum_{e \in R} c(e),\end{aligned}$$

where R is the set of k most expensive edges of the edge cut set $E(A, B)$. (We assume $\mathsf{Cost}_k^{\exp}(A, B) = 0$ whenever $|E(A, B)| \leq k$.) We consider the following variant of the classic MINCUT problem MIN s-t-CUT WITH k FREE EXPENSIVE EDGES: *Does there exist an s-t-cut (A, B) of G of discounted cost $\mathsf{Cost}_k^{\exp}(A, B) \leq \beta$, given two terminal vertices s and t, an integer $k \geq 0$, and the budget $\beta \geq 0$.* Let us note that for $k = 0$, MIN s-t-CUT$-k$ EXP is the problem of finding a minimum s-t-cut.

A close well-studied relative of the minimum s-t-cut problem is the GLOBAL MIN-CUT, or the edge connectivity of a graph. Here the task is to identify the set of edges of minimum weight in a connected graph whose removal disconnects the graph. We define the discounted version of GLOBAL MIN-CUT as follows, MINCUT WITH k FREE EXPENSIVE EDGES: Does there exist a cut (A, B) of G such that $\mathsf{Cost}_k^{\exp}(A, B) \leq \beta$, given an integer $k \geq 0$, and the budget $\beta \geq 0$. The problem MINCUT WITH k FREE CHEAP EDGES, or MINCUT$-k$ CHEAP for short, is defined similarly.

Another important cut problem is the MAXIMUM CUT problem, where the goal is to partition a graph to maximize the total weight of edges crossing the cut. This problem is significantly more challenging than the corresponding minimization variants. Analogously, we define four versions of the MAXIMUM CUT problem based on the presence or absence of terminals s and t, and the choice of discount function (excluding either cheap or expensive edges). In total, we thus obtain eight distinct discounted cut problems (see Table 1).

1.2 Our Results and Methods

In this section, we provide an overview of our main results. A central theme in our work is the use of a series of reductions that transform various discounted cut problems into their classic counterparts, enabling the application of classical techniques in new, cost-aware settings. Building on this, we provide complexity classification for all variants of the discounted cut problem. Together, these results offer a comprehensive view of the computational landscape for discounted cuts and reveal several surprising gaps in complexity between closely related problems. We summarize our results in Table 1.

Surprisingly, despite the long history of MIN s-t-CUT$-k$ EXP[1], we have not found an NP-hardness proof in the literature. The only known complexity lower bound is an NP-hardness result for a more general *network interdiction* model, in which each edge e has a resource cost $r(e)$. The goal in this model is to reduce the capacity of every s-t-cut by deleting a set of edges whose total resource cost does not exceed a threshold R. In this setting, MIN s-t-CUT$-k$ EXP is a special case where $r(e) = 1$ for every edge of G and $R = k$. A simple reduction from KNAPSACK shows that this more general interdiction problem is *weakly* NP-complete (Ball, Golden, and Vohra 1989; Wood 1993). Our first result significantly strengthens the known intractability bounds by establishing NP-completeness of MIN s-t-CUT$-k$ EXP when edge costs are restricted to one and two.

Theorem 1. MIN s-t-CUT$-k$ EXP *is NP-complete for instances where each edge has cost one or two. Moreover, the problem is W[1]-hard parameterized by k for instances with integer edge costs upper bounded by a polynomial of the size of the input graph.*

Due to space limitations, proofs for most theorems and lemmas have been omitted and are available in the appendix. Additionally, we have included a preliminaries section in the appendix, covering definitions and fundamental results in graph theory, complexity theory, and computational geometry.

While MIN s-t-CUT$-k$ EXP is NP-complete on general graphs, our next theorem establishes that on graphs of bounded genus, *all* eight variants of the discounted cut problem are solvable in polynomial time when the maximum edge cost is polynomial in n. The proof relies on a generic approach that reduces discounted cuts with polynomial weights to their classic counterparts on bounded-genus graphs. This reduction builds on an algebraic technique (Galluccio, Loebl, and Vondrák 2001) for handling generating functions of cuts in bounded-genus graphs.

Theorem 2. *Each of the eight problems*

- MIN/MAXCUT-k CHEAP/EXP,
- MIN/MAX s-t-CUT-k CHEAP/EXP

restricted to graphs of genus g, given with the corresponding embedding, admits a randomized algorithm of running time

$$\left(4^g \cdot n^{1.5} + m \cdot M\right) \cdot m^2 \cdot M \cdot \log^{\mathcal{O}(1)}(n + M),$$

[1] MIN s-t-CUT$-k$ EXP is also known as MOST VITAL LINKS.

	s-t-cut		Global cut	
	General	Planar	General	Planar
MIN EXP.	W[1]-hard (Thm 1)	P (Thm 4)	P (Thm 5)	P (Thm 5)
MAX CHEAP	paraNP-hard	P *(i)* (Thm 2)	paraNP-hard	P *(i)* (Thm 2)
MIN CHEAP	P (Theorem 3)	P (Thm 3)	P (Thm 3)	P (Thm 3)
MAX EXP.	paraNP-hard	P *(i)* (Thm 2)	paraNP-hard	P (Thm 3)

Table 1: Complexity classification of eight discounted-cut problems (minimum/maximum cut, s-t/global cut, k cheap/k expensive discounting), each studied on general and planar (bounded-genus) graphs, resulting in 16 complexity entries. The complexity is parameterized by k; entries marked "(i)" indicate tractability results known only for polynomial costs.

where M is the total edge cost. The algorithm can be derandomized for the cost of additional multiplier of order n in the running time.

We state Theorem 2 for all eight variants of discounted cuts. However, some of these problems admit polynomial-time algorithms on general graphs. The first observation is that the variants of "$\min -k$ cheap" and "$\max -k$ expensive" are not significantly different from their classic counterparts. Furthermore, this type of statement holds for optimization problems in the most general sense.

Consider the MINIMUM Π problem, whose task is, given a universe U, a cost function $c\colon U \to \mathbb{Z}_{\geq 0}$, and a family $\mathcal{F}$ of feasible subsets of U (which may be given implicitly), to find the minimum cost of a feasible subset. We define MINIMUM Π WITH k FREE CHEAP ELEMENTS as the problem whose input additionally includes an integer $k \geq 0$, and whose task is to find the minimum cost of a feasible subset, excluding the cost of the k cheapest elements. Similarly, we define MAXIMUM Π WITH k FREE EXPENSIVE ELEMENTS for the MAXIMUM Π problem, where the objective is to find the maximum cost of a feasible subset while excluding the cost of the k most expensive elements.

Theorem 3. *If* MINIMUM Π *(*MAXIMUM Π*, respectively) can be solved in $T(|U|, \|\mathcal{F}\|, M)$ time then* MINIMUM Π WITH k FREE CHEAP ELEMENTS *(*MAXIMUM Π WITH k FREE EXPENSIVE ELEMENTS*, respectively) can be solved in $\mathcal{O}(|U| \cdot T(|U|, \|\mathcal{F}\|, M))$ time where M is the maximum value of the cost function.*

Hence, if MINIMUM Π is solvable in polynomial time, then MINIMUM Π WITH k FREE CHEAP ELEMENTS is as well, and if MAXIMUM Π is, then so is MAXIMUM Π WITH k FREE EXPENSIVE ELEMENTS. By pipelining Theorem 3 with known complexity results for classic variants of cut problems, we obtain a number of corollaries.

Since MIN s-t-CUT is polynomial-time solvable in almost linear time (Chen et al. 2023), by Theorem 3, MIN s-t-CUT$-k$ CHEAP can be solved in $\mathcal{O}(m^{2+o(1)} \log M)$ time for instances with cost functions of maximum value M. Furthermore, this result holds for directed graphs.

Similarly, MINCUT can be solved in polynomial time by the randomized Karger's algorithm (Karger and Stein 1996) or the deterministic Stoer–Wagner algorithm (Stoer and Wagner 1997). In particular, using the improved version of Karger's algorithm given by Gawrychowski, Mozes, and Weimann (2024), we have that MINCUT$-k$ CHEAP is solvable in $\mathcal{O}(m^2 \log^2 n \log M)$ time by a randomized algorithm for instances with cost functions of maximum value M.

The classic MAXCUT is NP-complete. However, MAXCUT can be solved in polynomial time on planar graphs (Hadlock 1975), graphs excluding K_5 as a minor (Barahona 1983), and graphs of bounded genus (Galluccio, Loebl, and Vondrák 2001). Combining Theorem 3 and the results from (Liers and Pardella 2012), we obtain that MAXCUT$-k$ EXP can be solved in $\mathcal{O}(n^{5/2} \log n \log M)$ time on planar graphs for instances with cost functions of maximum value M.

By Theorem 2, MIN s-t-CUT$-k$ EXP is solvable in polynomial time on graphs of bounded genus with polynomial costs. We show that on planar graphs, the constraint on costs can be excluded. Interestingly, already the first papers (Wollmer 1963; McMasters and Mustin 1970) on MIN s-t-CUT$-k$ EXP studied planar graphs. However, polynomial time algorithms were known only for the situation when both terminal vertices s and t lie on outer face.

Theorem 4. *On planar graphs,* MIN s-t-CUT$-k$ EXP *can be solved in time $\mathcal{O}(kn^2 \log n \log M)$ where M is the maximum value of the cost function.*

The core idea in the proof of Theorem 4 is that computing a discounted cut in a planar graph G can elegantly be reduced to finding specific walks in its dual graph G^*. More specifically, every minimal cut in G corresponds to a cycle in G^*. Let us fix an s-t-path P in G. The key observation (sometimes known as the *discrete Jordan curve theorem*) is that a cycle in G^* corresponds to an s-t-cut in G if and only if it crosses P an odd number of times. Consequently, the problem reduces to finding a minimum-cost closed walk in G^* that intersects P an odd number of times, which can be solved using dynamic programming.

Finally, for MINCUT$-k$ EXP, we give a polynomial-time algorithm that works with arbitrary edge costs on general graphs. In light of the NP-hardness of MIN s-t-CUT WITH k FREE EXPENSIVE EDGES (see Theorem 1), this result highlights a surprising difference in the complexity of these two problems.

Theorem 5. MINCUT$-k$ EXP *admits a randomized algorithm that runs in time $\mathcal{O}(n^3 \cdot m \cdot \log^4 n \cdot \log\log n \cdot \log M)$ where M is the maximum value of the cost function.*

The proof of Theorem 5 relies on an algorithm for the BICRITERIA GLOBAL MINIMUM CUT. In this problem, we are given a graph G with two edge weight functions $w_1, w_2 \colon E(G) \to \mathbb{R}_{\geq 0}$ and real numbers $b_1, b_2 \geq 0$. The task is to decide whether there is a cut (A, B) of G such that $w_i(E(A, B)) \leq b_i$ for every $i \in \{1, 2\}$. By (Armon and Zwick 2006), this problem is solvable in polynomial time. (The algorithm of Armon and Zwick runs in polynomial time even in a more general setting for an arbitrary fixed number k of criteria. But for the proof of Theorem 5, we need only $k = 2$.) The fastest known algorithm for BICRITERIA GLOBAL MINIMUM CUT is due to Aissi, Mahjoub, and Ravi (2017), and provides a solution in time $\mathcal{O}(n^3 \log^4 n \log\log n)$ with probability $1 - 1/\Omega(n)$.

1.3 Related Work

The problem of determining the *most vital link* (singular) (Wollmer 1963) in a graph is the following. Given a flow network (G, s, t, c), that is, a graph G with source and target vertices, and a capacity function, determine which edge in G should be removed to obtain a graph with the smallest possible maximum s-t-flow. Clearly, this can be done in polynomial time, since we can try every edge and run a max-flow algorithm. By the Max Flow–Min Cut theorem, we want to find an edge whose removal reduces the Min s-t-cut as much as possible. Wollmer studied the problem on s-t-planar graphs, i.e., where the flow network is planar and s and t are both on the outer face. Using a correspondence between minimal cuts and paths in planar graphs, Wollmer provided a linear-time algorithm for the problem on this class of graphs.

Wollmer extended the concept of most vital link to the MOST VITAL LINKS problem (Wollmer 1964), where the objective is to remove k edges to minimize the s-t-flow. Again, by the Max Flow–Min Cut theorem, this is equivalent to deleting k edges to obtain a graph with as small a minimum s-t-cut as possible. Therefore, we want to find a minimal cut and delete edges only from that cut—k edges with highest capacities. The remaining edges will be the resulting cut. This implies that MOST VITAL LINKS is equivalent to MIN s-t-CUT$-k$ EXP.

Phillips (1993) studied this variant and provided a Polynomial-Time Approximation Scheme (PTAS) for planar graphs. Later, Wood (1993) claimed that the MOST VITAL LINKS on planar graphs had been solved. However, Wood's assertion merely referenced Wollmer's result, which applies exclusively to s-t-planar graphs. In fact, Wollmer's algorithm assumes s and t share a face in the planar embedding, as it starts by creating a direct edge between s and t, which is not feasible otherwise.

The MOST VITAL LINKS has been studied under various names across different fields. One such area is *graph vulnerability*, which examines how much a graph "breaks apart" when a certain number of edges or vertices are removed. Another is *network inhibition* (Phillips 1993) or *network interdiction* (Wood 1993), where the goal is to "damage" a graph as effectively as possible while using minimal resources. McMasters and Mustin (1970) study the problem of allocating a limited number of aircraft to interdict an enemy's supply lines on a particular day using and give a linear programming algorithm for the case where the cost of destroying an edge is the same as its capacity. Like Wollmer, they consider only s-t-planar graphs.

Later, several variations of the MOST VITAL LINKS problem were studied, mostly under the names NETWORK INTERDICTION and NETWORK INHIBITION. Many of these problems were modeled as Stackelberg games, in which an interdictor attempts to damage the network while a defender tries to repair or reinforce it (Steinrauf 1991). Zenklusen (2010) studied several interdiction problems, including flow interdiction on planar graphs and later, matching interdiction (Zenklusen 2014), i.e., removing edges to make the maximal matching as small as possible.

Interestingly, despite the MOST VITAL LINKS problem's long history and extensive bibliography, several open algorithmic questions remain about the minimum and maximum cuts whose cost is measured up to k edges. In this paper, we address these gaps and introduce additional algorithmic questions, hoping to stimulate further progress in this area.

2 General Technique for Subset Problems with Free Elements

In this section, we present general results for minimization (maximization, respectively) problems with free cheap (expensive, respectively) elements. We show that, given an algorithm $\mathcal{A}$ for MINIMUM Π (MAXIMUM Π, respectively), we can solve MINIMUM Π WITH k FREE CHEAP ELEMENTS (MAXIMUM Π WITH k FREE EXPENSIVE ELEMENTS, respectively) using at most $|U|$ calls of $\mathcal{A}$.

For an instance $(U, c, \mathcal{F})$ of MINIMUM Π (MAXIMUM Π, respectively), let $\mathsf{Opt}_{\min}(U, c, \mathcal{F})$ ($\mathsf{Opt}_{\max}(U, c, \mathcal{F})$, respectively) be the optimum cost of a feasible solution, and we use $\mathsf{Opt}^{\text{cheap}}_{\min}(U, c, \mathcal{F}, k)$ and $\mathsf{Opt}^{\exp}_{\max}(U, c, \mathcal{F}, k)$ for the optimum cost for MINIMUM Π WITH k FREE CHEAP ELEMENTS and MAXIMUM Π WITH k FREE EXPENSIVE ELEMENTS, respectively. We assume that these values are zeros if optimum subsets are of cardinality at most k. Let $c \colon U \to \mathbb{Z}_{\geq 0}$ be a function and let $w \geq 0$ be an integer. We use c_w and c^w for functions

$$c_w(x) = \begin{cases} c(x) & \text{if } x > w \\ w & \text{if } x \leq w \end{cases} \qquad c^w(x) = \begin{cases} c(x) & \text{if } x < w \\ w & \text{if } x \geq w \end{cases}$$

and write $C(X) = \{c(x) \mid x \in X\}$ for $X \subseteq U$.

Now, let $\mathrm{O}_{\min} = \min_{w \in C(U)}(\mathsf{Opt}_{\min}(U, c_w, \mathcal{F}) - kw)$ and $\mathrm{O}_{\max} = \max_{w \in C(U)}(\mathsf{Opt}_{\max}(U, c^w, \mathcal{F}) - kw)$

Lemma 1. *For an instance* $(U, c, \mathcal{F})$ *of* MINIMUM Π *(*MAXIMUM Π*, respectively) and* $k \geq 0$*,*

- $\mathsf{Opt}^{\text{cheap}}_{\min}(U, c, \mathcal{F}, k) = \max\{0, \mathrm{O}_{\min}\}$
- $\mathsf{Opt}^{\exp}_{\max}(U, c, \mathcal{F}, k) = \max\{0, \mathrm{O}_{\max}\}$.

Lemma 1 proves that MINIMUM Π WITH k FREE CHEAP ELEMENTS and MAXIMUM Π WITH k FREE EXPENSIVE ELEMENTS can be reduced to MINIMUM Π and MAXIMUM Π, respectively. This proves Theorem 3.

In particular, if MINIMUM Π or MAXIMUM Π can be solved in polynomial time then the corresponding problem with free edges admits a polynomial-time algorithm as well. It is well-known that MIN s-t-CUT is polynomial-time solvable and, by the recent results of Chen et al. (2023), the minimum s-t-cut can be found in an almost linear time. This implies the following proposition.

Proposition 1. MIN s-t-CUT$-k$ CHEAP *can be solved in* $\mathcal{O}(m^{2+o(1)} \log M)$ *time for instances with cost functions of maximum value* M*. Furthermore, this result holds for directed graphs.*

Similarly, MINCUT can be solved in polynomial time by the randomized Karger's algorithm (Karger and Stein 1996) or the deterministic Stoer–Wagner algorithm (Stoer and Wagner 1997). In particular, using the improved version of Karger's algorithm (Gawrychowski, Mozes, and Weimann 2024), we obtain the following proposition.

Proposition 2. MINCUT$-k$ CHEAP *can be solved in* $\mathcal{O}(m^2 \log^2 n \log M)$ *time by a randomized algorithm for instances with cost functions of maximum value* M*.*

Finally in this section, we note that MAXCUT can be solved in polynomial time on planar graphs (Hadlock 1975), graphs excluding K_5 as a minor (Barahona 1983), and graphs of bounded genus (Galluccio, Loebl, and Vondrák 2001). We discuss this problem in detail in the next section. Here, we only mention that combining Theorem 3 and the results of Liers and Pardella (Liers and Pardella 2012), we obtain the following result for planar graphs.

Proposition 3. MAXCUT$-k$ EXP *can be solved in* $\mathcal{O}(n^{5/2} \log n \log M)$ *time on planar graphs for instances with cost functions of maximum value* M*.*

2.1 Lower Bounds for MIN s-t-CUT WITH k FREE EXPENSIVE EDGES

In this section we prove that MIN s-t-CUT WITH k FREE CHEAP EDGES is NP-complete and W-hard even in very restricted cases. Notice that the problem can be solved in polynomial time when the edges of the input graph have unit costs—to obtain an optimum solution, it is sufficient to find the minimum s-t-cut in the considered graph and subtract k from its size. We prove that the problem becomes NP-hard when we allow edges to have cost 1 or 2 and show that MIN s-t-CUT$-k$ EXP is W[1]-hard when parameterized by k, the number of edges to delete, when we allow polynomial (in the graph size) costs.

Theorem 1. MIN s-t-CUT$-k$ EXP *is NP-complete for instances where each edge has cost one or two. Moreover, the problem is W[1]-hard parameterized by k for instances with integer edge costs upper bounded by a polynomial of the size of the input graph.*

3 Graphs of Bounded Genus

In this section, we prove that all eight variants of the *cuts with k free edges* problems can be solved in polynomial time in graphs of bounded genus, when restricted to polynomially-bounded integer edge costs. That is, our algorithms depend on M polynomially, where M is the total edge cost. We remark that our algorithms require an embedding of G into an orientable surface of genus g given as an input; such embeddings are known to be computable in time $2^{g^{\mathcal{O}(1)}} n$ (Mohar 1999; Kawarabayashi, Mohar, and Reed 2008).

We rely heavily on the work of Galluccio, Loebl, and Vondrák (2001). They show that integral total cut costs can be computed efficiently—through polynomial evaluation and interpolation—once the corresponding embedding is known. We outline their approach in the proof of the following lemma.

Algorithm 1: Algorithm resolving an instance (G, c, k, β) of a problem Π. If Π is s-t-CUT, s and t are also a part of the input. Additionally, an embedding ϕ of G into an oriented surface of genus g is given.

```
e_1, e_2, ..., e_m ← ordering of E(G) s.t. c(e_i) ≤ c(e_{i+1})
M ← 1 + \sum_{i=1}^{m} c(e_i)
for t ∈ [m + 1] do
    G', φ' ← copy of G, φ
    for i ∈ [m] do
        if i < t and Π is FREE CHEAP or
        if i ≥ t and Π is FREE EXPENSIVE then
            c'(e_i) ← M
        else
            c'(e_i) ← c(e_i)
        end if
        if Π is s-t-CUT then
            add edge e_{m+1} between s and t in G'
            attach a new handle between s and t in φ' and
            embed e_{m+1} in it in φ'
            c'(e_{m+1}) ← m · M + 1
            T ← c'(e_{m+1})
        else
            T ← 0
        end if
    end for
    C ← output of the algorithm of Lemma 2 applied to
    G', c' and φ'
    for w' ∈ {w − T : w ∈ C, w ≥ T} do
        β' ← w' (mod M) {reimbursed cost}
        k' ← ⌊w'/M⌋ {number of reimbursed edges}
        if β' ≤ β and k' ≤ k and Π is MIN or
        if β' ≥ β and k' ≥ k and Π is MAX then
            return YES
        end if
    end for
end for
if β = 0 and Π is MAX then
    return YES
end if
return NO
```

Lemma 2. *There is a randomized algorithm that, given an* n*-vertex graph* G *with positive integral edge costs, along with its embedding* ϕ *in an orientable surface of genus* g*, in time* $(4^g \cdot M \cdot n^{1.5} + M^2) \cdot \log^{\mathcal{O}(1)}(n + M)$*, where* M *is the total edge cost, computes a set* $C \subseteq \{0, 1, \ldots, M\}$*, that satisfies*

- *if $w \in C$, then there is a cut of cost w in G;*
- *if there is a cut of cost w in G, then $w \in C$ with probability at least $\frac{1}{2}$.*

The algorithm can be derandomized for the cost of additional multiplier of order n in the running time.

Proof sketch. We encode all possible cut costs of G into the polynomial $\mathcal{C}(G,x) = \sum_{A \cup B = V(G)} x^{c(E(A,B))} = \sum_{w=0}^{M} a_w x^w$, where $a_w \neq 0$ iff G has a cut of cost w. Using the algorithm of (Galluccio, Loebl, and Vondrák 2001), we can evaluate $\mathcal{C}(G,i) \bmod p$ for any integer i in time $4^g \cdot n^{1.5} \cdot \log^{\mathcal{O}(1)}(n+M)$ given a genus-g embedding. The algorithm picks a random prime p from $2n$ primes larger than M, evaluates $\mathcal{C}(G,i) \bmod p$ for all $i \in \{0,\dots,M\}$, and recovers the coefficients modulo p using Lagrange interpolation. The set of candidate cut costs C_p is then all w for which the recovered coefficient is nonzero.

Correctness follows since any true cut cost w appears in C_p unless the random prime p divides the coefficient a_w. As $a_w \leq 2^n$, at most n primes can divide it, giving success probability at least $1/2$. The total running time is $(4^g \cdot M \cdot n^{1.5} + M^2) \cdot \log^{\mathcal{O}(1)}(n+M)$, dominated by the evaluation and interpolation phases. Derandomization is achieved by testing $n+1$ distinct primes and returning the union of all resulting sets C_p. ☐

The main result of this section is the following, restating from the introduction, whose formal proof can be found in the appendix.

Theorem 2. *Each of the eight problems*

- MIN/MAXCUT-k CHEAP/EXP*,*
- MIN/MAX s-t-CUT-k CHEAP/EXP

restricted to graphs of genus g, given with the corresponding embedding, admits a randomized algorithm of running time

$$\left(4^g \cdot n^{1.5} + m \cdot M\right) \cdot m^2 \cdot M \cdot \log^{\mathcal{O}(1)}(n+M),$$

where M is the total edge cost. The algorithm can be derandomized for the cost of additional multiplier of order n in the running time.

While Theorem 2 asserts that the eight problems admit polynomial-time algorithms, these algorithms share similarities and adhere to a common framework. The detailed proof can be found in the appendix, where we present a general algorithmic framework from which all eight algorithms follow as specific instantiations.

3.1 Planar MIN s-t-CUT WITH k FREE EXPENSIVE EDGES

Now we prove Theorem 4 which shows that MIN s-t-CUT$-k$ EXP is solvable in polynomial time on planar graphs with exponential edge costs. Throughout the section, we assume that all graphs considered are plane, meaning they come equipped with an embedding as the planarity can be tested and a planar embedding can be found (if it exists) in linear time, as shown by Hopcroft and Tarjan (1974).

The crucial idea behind our algorithm is to switch to the dual problem. Recall that (A,B) is a minimal s-t-cut in a connected graph G if and only if $C = \{e^* \in E(G^*) \mid e \in E(A,B)\}$ is a cycle of G^* such that s and t are in distinct faces of C. Observe that given an instance (G,c,s,t,k,β) of MIN s-t-CUT$-k$ EXP, an s-t-cut (A,B) of minimum discounted cost $\mathsf{Cost}_k^{\exp}(A,B)$ should be minimal. Thus, the problem reduces to finding a cycle C in G^* of minimum discounted cost such that s and t are in distinct faces of C. Our first task is to establish a criteria that guarantees that s and t are in distinct faces of a cycle of G^*.

Let G be a plane graph. We say that a path P in G *crosses* a cycle C^* of G^* in $e \in E(P)$ if C^* contains the edge $e^* \in E(G^*)$ that is dual to e. The *number of crosses* of P and C^* is the number of edges of P where P and C^* cross. We can make the following observation.

Observation 1 (Bentert et al. (2024))**.** *Let G be a plane graph, let $s,t \in V(G)$, and let P be an s-t-path. For any cycle C^* of G^*, s and t are in distinct faces of C^* if and only if the number of crosses of P and C^* is odd.*

Thus, to solve MIN s-t-CUT$-k$ EXP for (G,c,s,t,k,β), we have to find a cycle C in G^* of minimum discounted cost with odd number of crosses with some s-t-path in G. To solve this problem, we use the polynomial Dijkstra-style algorithm finding an odd s-t-walk of minimum discounted cost in a general graph. Let G be a graph with a given cost function $c\colon E(G) \to \mathbb{Z}_{\geq 0}$. Let also $k \geq 0$ be an integer. For a walk $W = v_0 \dots v_\ell$, let R be the sequence of k most expensive edges $v_{i-1}v_i$ in W; note that R may include several copies of the same edge of G. Then we define $\mathsf{Cost}_k^{\exp}(W) = \sum_{i=1}^{\ell} c(v_{i-1}v_i) - \sum_{e \in R} c(e)$.

We note that the following lemma holds for all graphs:

Lemma 3. *Given a graph G with a cost function $c\colon E(G) \to \mathbb{Z}_{\geq 0}$, two vertices s and t, and an integer $k \geq 0$, an odd s-t-walk W with minimum $\mathsf{Cost}_k^{\exp}(W)$ can be found in $\mathcal{O}(mk \log n \log M)$ time where M is the maximum value of the cost function.*

We also use the following folklore observation.

Observation 2. *If $G = (V,E)$ contains an odd closed walk $W = v_0 \dots v_\ell$, then it also contains an odd cycle C that is part of W, that is, $V(C) \subseteq \{v_0,\dots,v_\ell\}$ and for every $e \in E(C)$, there is $i \in [\ell]$ such that $e = v_{i-1}v_i$.*

Now we are ready to prove Theorem 4 which we restate:

Theorem 4. *On planar graphs,* MIN s-t-CUT$-k$ EXP *can be solved in time $\mathcal{O}(kn^2 \log n \log M)$ where M is the maximum value of the cost function.*

Proof. Let G be a plane graph with a cost function $c\colon E(G) \to \mathbb{Z}_{\geq 0}$, and let s and t be two distinct vertices, and $k \geq 0$ be an integer. We assume that G is connected as MIN s-t-CUT WITH k FREE EXPENSIVE EDGES is trivial if s and t are in distinct connected components.

We find an arbitrary s-t-path P in G. Then we construct the dual graph G^* with the corresponding cost function c^*. For each edge $e^* = uv \in E(G^*)$ such that the dual edge $e \notin E(P)$, we subdivide this edge, that is, we introduce a new vertex w and replace uv with uw and wv. Denote the obtained graph $G^\star$. We define the cost function $c^\star\colon E(G^\star) \to \mathbb{Z}_{\geq 0}$ by setting $c^\star(e^*) = c^*(e^*) = c(e)$ for

Algorithm 2: Cheapest odd walk with k free edges

Create a priority queue Q
Q.push$(0, 0, 0, s)$
$\text{dist}(v, k, p) \leftarrow \infty$ {all values initialized to ∞}
while Q $\neq \emptyset$ **do**
  $(\ell, k', p, v) \leftarrow$ Q.pop()
  if $\text{dist}(k', p, v) \leq \ell$ **then**
    continue {discard sub-optimal vector}
  end if
  $\text{dist}(k', p, v) \leftarrow \ell$
  for $u \in N_G(v)$ **do**
    Q.push$(\ell + w(u, v), k', \overline{p}, u)$
    if $k' < k$ **then**
      Q.push$(\ell, k' + 1, \overline{p}, u)$
    end if
  end for
end while

$e \in E(P)$, and we set $c^\star(uw) = c^*(uv)$ and $c^\star(wv) = 0$ if uw and wv were obtained by subdividing $e^* = uv$. The construction of $G^\star$ and the definition of the cost function $c^\star$ immediately imply the following property.

Claim 1. *The graph G^* has a cycle C^* with odd number of crossing of P if and only if $G^\star$ has an odd cycle $C^\star$. Moreover, $\min \mathsf{Cost}_k^{\exp}(C^*)$ taken over all cycles C^* with odd number of crossing of P is the same as $\min \mathsf{Cost}_k^{\exp}(C^\star)$ where the minimum is taken over all odd cycles $C^\star$ in $G^\star$.*

Thus, we reduced our initial task to finding an odd cycle in $G^\star$ of minimum discounted cost. To find such a cycle, we use the algorithm from Lemma 3. We go trough every vertex $x \in V(G^*)$, and run the algorithm to find an odd x-x-walk W with minimum $\mathsf{Cost}_k^{\exp}(W)$. Among all such walks, we find a closed walk $W^\star$ of minimum discounted cost. By Observation 2, there is an odd cycle $C^\star$ that is a part of $W^\star$. Notice that $\mathsf{Cost}_k^{\exp}(C^\star) \leq \mathsf{Cost}_k^{\exp}(W^\star)$. Because $W^\star$ is an odd closed walk of minimum discounted cost, we obtain that $C^\star$ is an odd cycle in $G^\star$ of minimum discounted cost $\mathsf{Cost}_k^{\exp}(C^\star) = \mathsf{Cost}_k^{\exp}(W^\star)$. Since the minimum value of $\mathsf{Cost}_k^{\exp}(A, B)$ of an s-t-cut (A, B) of G is the same as $\mathsf{Cost}_k^{\exp}(C^\star)$, we return the cut corresponding to the cycle $C^\star$ in $G^\star$. This concludes the description of the algorithm and its correctness proof. It is easy to verify that the running is $\mathcal{O}(kn^2 \log n \log M)$.

□

4 Global MINCUT WITH k FREE EXPENSIVE EDGES

This section is devoted to the *global* version of the minimum cut problem with reimbursed edges, namely MINCUT$-k$ EXP. The problem is well-known for the case $k = 0$, and the state-of-the-art randomized algorithm is due to Gawrychowski, Mozes, and Weimann (Gawrychowski, Mozes, and Weimann 2024) and runs in $\mathcal{O}(m \cdot \log^2 n \cdot \log M)$ time, where M is the maximum edge cost value.

The *multi-criteria* global minimum cut problem, where edges have multiple weight types that are measured separately, was introduced by Armon and Zwick (Armon and Zwick 2006) and later improved in (Aissi, Mahjoub, and Ravi 2017). In this work, we focus on the two-criteria case and observe that MINCUT$-k$ EXP is a special case of the BICRITERIA GLOBAL MINIMUM CUT problem: Given a graph with two edge weight functions w_1 and w_2, does there exist a cut (A, B) of G such that $w_i(E(A, B)) \leq b_i$ for every $i \in \{1, 2\}$?

Proposition 4 (Aissi, Mahjoub, and Ravi (2017))**.** BICRITERIA GLOBAL MINIMUM CUT *admits a randomized algorithm with $\mathcal{O}(n^3 \cdot \log^4 n \cdot \log\log n \cdot \log M)$ running time, where M is the maximum edge cost value.*

We use Proposition 4 to prove Theorem 5. The basic idea is to consider $m + 1$ choices of splitting edges into *cheap* and *expensive* and construct an instance of BICRITERIA GLOBAL MINIMUM CUT for each choice. Before giving the proof, we restate Theorem 5 for the reader's convenience.

Theorem 5. MINCUT$-k$ EXP *admits a randomized algorithm that runs in time $\mathcal{O}(n^3 \cdot m \cdot \log^4 n \cdot \log\log n \cdot \log M)$ where M is the maximum value of the cost function.*

Proof Sketch. The algorithm sorts edges by non-decreasing cost and, for each threshold $t \in \{1, \ldots, m+1\}$, constructs a BICRITERIA GLOBAL MINIMUM CUT instance where edges cheaper than the t-th edge contribute to the cost function w_1, and edges from the t-th onward are treated as potentially free and counted under w_2. The original instance of MINCUT$-k$ EXP is then a yes-instance if and only if one of these $m + 1$ bicriteria instances admits a cut of total w_1-cost at most W and w_2-count at most k. Each instance is solved using the randomized algorithm of Proposition 4, giving a total running time of $\mathcal{O}(n^3 m \log^4 n \log\log n \log M)$.

Correctness follows by mapping any feasible cut of MINCUT$-k$ EXP to the threshold t corresponding to its cheapest reimbursed edge (or $m + 1$ if none are free). In that thresholded bicriteria instance, the same cut satisfies both cost and free-edge constraints. Conversely, any feasible solution to a bicriteria instance respects the ordering of cheap and free edges and thus yields a valid cut for MINCUT$-k$ EXP. □

5 Conclusion

We initiated a comprehensive study of cut problems with discounts and conclude with three open problems. We showed that on general graphs, MIN s-t-CUT$-k$ EXP is W[1]-hard parameterized by k for instances with integer edge costs upper bounded by a polynomial of the size of the input graph. Does the problem become fixed-parameter tractable FPT when the edge costs are constants? Is the problem FPT when parameterized by combined parameters k and the maximum edge cost? Our polynomial time algorithm for MAXCUT$-k$ CHEAP on graphs of bounded genus relies on the assumption that the costs are polynomial. Is the problem polynomial time solvable on planar graphs for arbitrary weights?

Acknowledgments

Danil Sagunov: This work was supported by the Ministry of Economic Development of the Russian Federation (IGK 000000C313925P4C0002), agreement №139-15-2025-010.

References

Aissi, H.; Mahjoub, A. R.; and Ravi, R. 2017. Randomized Contractions for Multiobjective Minimum Cuts. In *25th Annual European Symposium on Algorithms (ESA)*, volume 87 of *LIPIcs*, 6:1–6:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Armon, A.; and Zwick, U. 2006. Multicriteria Global Minimum Cuts. *Algorithmica*, 46(1): 15–26.

Ball, M. O.; Golden, B. L.; and Vohra, R. V. 1989. Finding the most vital arcs in a network. *Operations Research Letters*, 8(2): 73–76.

Barahona, F. 1983. The max-cut problem on graphs not contractible to K_5. *Operations Research Letters*, 2(3): 107–111.

Bentert, M.; Drange, P. G.; Fomin, F. V.; Golovach, P. A.; and Korhonen, T. 2024. Two-Sets Cut-Uncut on Planar Graphs. In *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia*, volume 297 of *LIPIcs*, 22:1–22:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Chen, L.; Kyng, R.; Liu, Y. P.; Peng, R.; Gutenberg, M. P.; and Sachdeva, S. 2023. Almost-Linear-Time Algorithms for Maximum Flow and Minimum-Cost Flow. *Commun. ACM*, 66(12): 85–92.

Galluccio, A.; Loebl, M.; and Vondrák, J. 2001. Optimization via enumeration: a new algorithm for the max cut problem. *Mathematical Programming*, 90: 273–290.

Gawrychowski, P.; Mozes, S.; and Weimann, O. 2024. Minimum Cut in $O(m \log^2 n)$ Time. *Theory Comput. Syst.*, 68(4): 814–834.

Hadlock, F. 1975. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4(3): 221–225.

Hopcroft, J. E.; and Tarjan, R. E. 1974. Efficient Planarity Testing. *J. ACM*, 21(4): 549–568.

Karger, D. R.; and Stein, C. 1996. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4): 601–640.

Kawarabayashi, K.; Mohar, B.; and Reed, B. A. 2008. A Simpler Linear Time Algorithm for Embedding Graphs into an Arbitrary Surface and the Genus of Graphs of Bounded Tree-Width. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 771–780. IEEE Computer Society.

Liers, F.; and Pardella, G. 2012. Partitioning planar graphs: A fast combinatorial approach for max-cut. *Comput. Optim. Appl.*, 51(1): 323–344.

McMasters, A. W.; and Mustin, T. M. 1970. Optimal interdiction of a supply network. *Naval Research Logistics Quarterly*, 17(3): 261–268.

Mohar, B. 1999. A Linear Time Algorithm for Embedding Graphs in an Arbitrary Surface. *SIAM Journal on Discrete Mathematics*, 12(1): 6–26.

Phillips, C. A. 1993. The network inhibition problem. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, 776–785.

Steinrauf, R. L. 1991. *Network Interdiction Models*. Master's thesis, Naval Postgraduate School Monterey Ca, Monterey, California.

Stoer, M.; and Wagner, F. 1997. A simple min-cut algorithm. *Journal of the ACM (JACM)*, 44(4): 585–591.

Wollmer, R. D. 1963. Some Methods for Determining the Most Vital Link in a Railway Network. Technical report, RAND Corporation.

Wollmer, R. D. 1964. Removing Arcs from a Network. *Operations Research*, 12(6): 934–940.

Wood, R. K. 1993. Deterministic network interdiction. *Mathematical and Computer Modelling*, 17(2): 1–18.

Zenklusen, R. 2010. Network flow interdiction on planar graphs. *Discrete Applied Mathematics*, 158(13): 1441–1455.

Zenklusen, R. 2014. Connectivity interdiction. *Oper. Res. Lett.*, 42(6-7): 450–454.