A Strongly Polynomial Time Algorithm for Multicriteria Global Minimum Cuts

Hassene Aissi[1], A. Ridha Mahjoub[1], S. Thomas McCormick[2], and Maurice Queyranne[2]

[1] PSL, Université Paris-Dauphine, LAMSADE, France
{aissi,mahjoub}@lamsade.dauphine.fr
[2] Sauder School of Business, University of British Columbia, Canada
{tom.mccormick,maurice.queyranne}@sauder.ubc.ca

Abstract. We investigate the bicriteria global minimum cut problem where each edge is evaluated by two nonnegative cost functions. The *parametric complexity* of such a problem is the number of linear segments in the parametric curve when we take all convex combinations of the criteria. We prove that the parametric complexity of the global minimum cut problem is $O(|V|^3)$. As a consequence, we show that the number of non-dominated points is $O(|V|^7)$ and give the first strongly polynomial time algorithm to compute these points. These results improve on significantly the super-polynomial bound on the parametric complexity given by Mulmuley [11], and the pseudo-polynomial time algorithm of Armon and Zwick [1] to solve this bicriteria problem. We extend some of these results to arbitrary cost functions and more than two criteria, and to global minimum cuts in hypergraphs.

1 Introduction

We consider the multicriteria version of the global minimum cut problem in undirected graphs. This problem is extensively studied in combinatorial optimization since many practical problems in, e.g., communications and electrical networks, contain it as a subproblem [1]. Let $G = (V, E)$ be an undirected graph, and $c^1, \dots, c^k : E \to \mathbb{R}_+$ be k nonnegative cost functions, or *criteria*, defined on its edges. A *cut* C of G is a subset $C \subseteq V$ such that $\emptyset \neq C \neq V$, and it contains the set of edges $\delta(C)$ with exactly one end in C. The *cost* of cut C w.r.t. criterion j is $c^j(C) \equiv c^j(\delta(C))$. We would like a cut that simultaneously minimizes all criteria, but such a solution usually does not exist.

Therefore, we focus on *Pareto optimal solutions*, i.e., solutions that cannot be improved upon in any criterion without degrading another criterion. Each cut C is associated with its criteria vector (or point) $y(C) = (c^1(C), \dots, c^k(C))$ in the criteria space $\mathbb{R}^k$. Let $Y = \{y(C) : \emptyset \neq C \subset V\}$ be the set of all criteria points associated with cuts (note that different cuts might give rise to the same criteria point). Given points $y(C), y(C') \in Y$, $y(C')$ *dominates* $y(C)$ if $y(C')_i \leqslant y(C)_i$, for all $i = 1, \dots, k$, and $y(C')_j < y(C)_j$ for at least one j. If there exists no

J. Lee and J. Vygen (Eds.): IPCO 2014, LNCS 8494, pp. 25–36, 2014.

$y(C') \in Y$ that dominates $y(C)$, then $y(C)$ is *non-dominated.* Let Y_{ND} be the set of non-dominated points in Y.

A vector of multipliers $\mu \in \mathbb{R}^k$ forms a convex combination if $\mu \geq 0$ and $\sum_{i=1}^{k} \mu_i = 1$; the set of all such multipliers is the *simplex* S^k. Given $y(C') \in Y_{ND}$, if there exists $\mu \in S^k$ such that $C' \in \arg\min_C\{\sum_{i=1}^{k} \mu_i c^i(C) : \emptyset \neq C \subset V\}$ then $y(C')$ is called a *Supported Non-Dominated* (SND) point. The non-dominated points that are not SND points are called *Unsupported Non-Dominated* (UND) points. By "solving" a multicriteria discrete optimization problem we mean generating all SND and UND points.

The computation of SND points is related to the field of *parametric optimization.* The function $f : S^k \mapsto \mathbb{R}$ defined by $f(\mu) = \min_C\{\sum_{i=1}^{k} \mu_i c^i(C) : \emptyset \neq C \subset V\}$ is piecewise linear and concave; the facets of its graph correspond to SND points [3]. The *parametric complexity* of a multicriteria problem is the maximum number of facets. Our main interest here is to study the parametric complexity of global minimum cut, mainly for the case where $k = 2$.

A natural subproblem of parametric minimum cut is solving single-criterion (ordinary) minimum cut, e.g., for some fixed value of μ. The fastest deterministic algorithms for this problem run in $O(|E| \cdot |V| + |V|^2 \log |V|)$ time (Nagamochi and Ibaraki [13] and Stoer and Wagner [19]). The fastest randomized algorithm runs in $O(|E| \log^3 |V|)$ time (Karger [7]). These algorithms are faster than minimum s–t-cut algorithms that are based on network flows. See [14] for a detailed treatment of graph connectivity problems.

The multicriteria versions of several combinatorial optimization problems has been extensively studied (see Ehrgott [3]). These problems are often *intractable* in the sense that the cardinality of the set of (supported) non-dominated points grows exponentially in the input size. Furthermore, it is often hard even to verify if a given point is non-dominated. Multicriteria global minimum cut is an exception to the above intractability results. Indeed, Armon and Zwick [1] show that the decision version of the global multicriteria problem can be solved in polynomial time. The proof relies on the fact that the single-criterion global minimum cut problem has at most a strongly polynomial number of near-optimal solutions. More precisely, given $\alpha \geqslant 1$, a cut is called *α-approximate* if its cost is less than α times the minimum. Karger and Stein [8] show that there are $O(|V|^{2\alpha})$ α-approximate cuts, and they give a randomized algorithm for finding them in $\widetilde{O}(|V|^{2\alpha})$ time. Nagamochi et al. [16] give a deterministic algorithm to find them in $O(|E||V|^{2\alpha})$ time, and they prove that there are $O(|V|^2)$ $\frac{4}{3}$-approximate cuts. Henzinger and Williamson [4] improve on this result by proving that there are $O(|V|^2)$ $\frac{3}{2}$-approximate cuts; they also show that $\frac{3}{2}$ is the largest possible approximation factor α for which there exist $O(|V|^2)$ α-approximate cuts.

For multicriteria global minimum cut, Armon and Zwick [1] used the result of [8] to give a pseudo-polynomial time algorithm to compute all the non-dominated points. Carstensen [2] shows that the parametric complexity of the s–t minimum cut problem is exponential for one parameter. Mulmuley [11] gives a simpler proof of this result, and studies the parametric complexity of the global minimum cut problem for $k = 2$. He considers the case where i) cost functions c^1 and

c^2 may be negative, ii) the parametric edge costs are positive and iii) the bit size of the values of c^1 and c^2 are at most a polynomial in $|V|$. And he shows in Theorem 3.10 that the parametric complexity is polynomial in this case. However, if iii) is relaxed, the proof of his theorem implies that the parametric complexity is $O(|V|^{19} \log |V| \log C_{max})$, where C_{max} is the maximum cost over all edges. This is surprising since the parametric function f is the minimum of the parametric functions of $O(|V|)$ minimum s–t cut problems (by fixing s and letting t vary over the other nodes), each of which could have an exponential number of breakpoints.

In this paper we give a much smaller, strongly polynomial upper bound on the parametric complexity of minimum cut, which leads to a strongly polynomial time algorithm for parametric global minimum cut, and hence a strongly polynomial time algorithm for the multicriteria version. In Section 2 we study in detail the bicriteria case, $k = 2$. In Section 3 we consider extensions, including arbitrary cost functions and more than two criteria, and global cuts in hypergraphs.

2 Complexity and Algorithms for $k = 2$

2.1 Parametric Complexity of the Global Min Cut Problem

We are given a graph $G = (V, E)$, and two nonnegative cost functions $c^1, c^2 : E \to \mathbb{R}_+$. For $\mu \in [0, 1]$, define the parametric cost function $c_\mu = \mu c^1 + (1-\mu)c^2$. Let $\mathcal{S}(G)$ denote the set of cuts which are optimal solutions for some $\mu \in [0, 1]$. Our main result is a $O(|V|^3)$ upper bound on $\mathcal{S}(G)$. For every cut $X \in \mathcal{S}(G)$, let $I(G, X)$ denote the largest sub-interval of $[0, 1]$ such that X is optimal for all $\mu \in I(G, X)$.

Theorem 1 *Assume that $\mu c^1(X) + (1 - \mu)c^2(X) > 0$ for every $X \in \mathcal{S}(G)$ and $\mu \in I(G, X)$. Then the parametric complexity of the global min cut problem is $O(|V|^3)$.*

The proof of Theorem 1 is non-constructive and relies on the following definitions. Let $H = (W, F)$ denote a graph (which may be a subgraph of G); $c^1, c^2 : F \to \mathbb{R}_+$ two nonnegative edge cost functions; and X a cut in H. If lines $\mu c^1(X) + (1 - \mu)c^2(X)$ and $\frac{\mu c^1(F)}{|W|} + \frac{(1-\mu)c^2(F)}{|W|}$ intersect in $[0, 1]$, let $INT(H, X) \in [0, 1]$ denote their intersection point. For optimal $X \in \mathcal{S}(H)$, let $I'(H, X) \subseteq I(H, X)$ be a maximal subinterval satisfying

$$\frac{\mu c^1(F)}{|W|} + \frac{(1-\mu)c^2(F)}{|W|} \leqslant \mu c^1(X) + (1 - \mu)c^2(X), \text{ for every } \mu \in I'(H, X). \quad (1)$$

Note that $I'(H, X)$ might be empty. Let $\mathcal{S}^{\geqslant}(H)$ denote the set of optimal solutions satisfying (1); $\mathcal{S}^{\geqslant}(H)$ might also be empty.

The set of solutions $\mathcal{S}(H) \setminus \mathcal{S}^{\geqslant}(H)$ can be partitioned into three subsets:

1. $\mathcal{S}_1^{<}(H) = \{X \in \mathcal{S}(H) \setminus \mathcal{S}^{\geqslant}(H)\colon I(H,X) \subseteq [0, INT(H,X)]\}$,
2. $\mathcal{S}_2^{<}(H) = \{X \in \mathcal{S}(H) \setminus \mathcal{S}^{\geqslant}(H)\colon I(H,X) \subseteq [INT(H,X), 1]\}$,
3. $\mathcal{S}_3^{<}(H) = \{X \in \mathcal{S}(H) \setminus \mathcal{S}^{\geqslant}(H)$: function $\mu c^1(X) + (1-\mu)c^2(X)$ is below $\frac{\mu c^1(F)}{|W|} + \frac{(1-\mu)c^2(F)}{|W|}$ in $[0,1]\}$.

Figure 1 depicts an example of function f having six facets associated to optimal solutions $\mathcal{S}(H) = \{X_1, \ldots, X_6\}$. Parametric function $\frac{\mu c^1(F)}{|W|} + \frac{(1-\mu)c^2(F)}{|W|}$ intersects the facets of f corresponding to X_2 and X_4. For X_4, for instance, we have $I(H,X_4) = [\mu_1, \mu_3]$, $INT(H,X_4) = \mu_2$, and $I'(H,X_4) = [\mu_1, \mu_2]$. However, for X_3 we have $I(H,X_3) = I'(H,X_3)$. Here we have, $\mathcal{S}^{\geqslant}(H) = \{X_2, X_3, X_4\}$, $\mathcal{S}_1^{<}(H) = \{X_1\}$, $\mathcal{S}_2^{<}(H) = \{X_5, X_6\}$ and $\mathcal{S}_3^{<}(H) = \emptyset$.

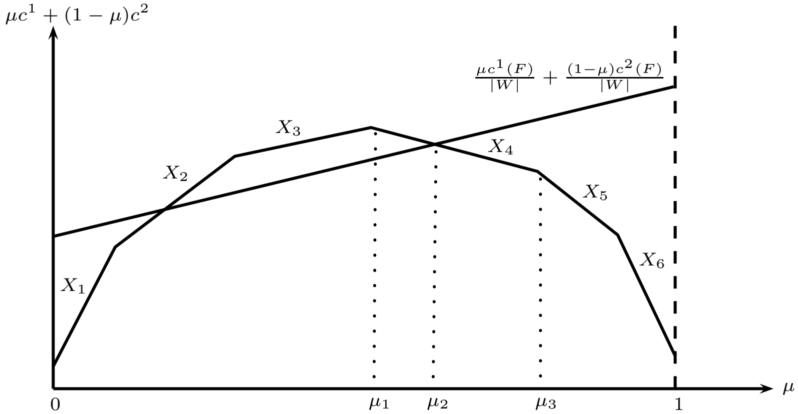

Fig. 1. Functions $f(\mu)$ and $\frac{\mu c^1(F)}{|W|} + \frac{(1-\mu)c^2(F)}{|W|}$

The proof of Theorem 1 uses the following lemma, whose proof is omitted.

Lemma 1 $|\mathcal{S}^{\geqslant}(H)| = O(|W|^2)$.

Proof. (of Theorem 1)

It suffices by Lemma 1 to give an upper bound on the cardinality of $\mathcal{S}_1^{<}(G) \cup \mathcal{S}_2^{<}(G) \cup \mathcal{S}_3^{<}(G)$. We focus on $\mathcal{S}_0 = \mathcal{S}_2^{<}(G) \cup \mathcal{S}_3^{<}(G)$ and show that its cardinality is $O(|V^3|)$. Set $\mathcal{S}_1^{<}(G)$ can be handled similarly. Assume that we have an oracle **O** that computes $\mathcal{S}(H)$ for any graph H.

In what follows we proceed in two steps in order to show that $|\mathcal{S}_0| \leqslant O(|V|^3)$. We will show that there exist two subsets $\mathcal{S}$ and $\mathcal{S}'$ such that $\mathcal{S}_0 \subseteq \mathcal{S} \cup \mathcal{S}'$, and $O(|\mathcal{S}|) \leqslant |V|^3$ and $O(|\mathcal{S}'|) \leqslant |V|$. This is a consequence of Algorithms 1 and 2 given below.

Figure 2 depicts the behavior of Algorithm 1. In each iteration r of this algorithm, an edge with large c^1-cost is contracted. Let $G_r = (V_r, E_r)$ denote the resulting graph obtained at iteration r. Note that the loops that arise by the contractions are kept in G_r. The idea is to have $|E_r| = |E_{r-1}| - 1$, and to ensure that $c^1(E_r)$ is not too small in comparison with $c^1(E_{r-1})$.

Consider the first iteration of the repeat loop. We will show next that once an edge is contracted in Step 4, $\mathcal{S}_0$ is partitioned into $\mathcal{S}^{\geqslant}(G_1)$, $\mathcal{S}_1^{<}(G_1)$ and $\mathcal{S}_2^{<}(G_1) \cup \mathcal{S}_3^{<}(G_1)$. By Lemma 1, we know that $|\mathcal{S}^{\geqslant}(G_1)| \leqslant O(|V_1|^2)$. We will show that the cardinality of $\mathcal{S}_1^{<}(G_1) \cap \mathcal{S}_0$ is also $O(|V_1|^2)$.

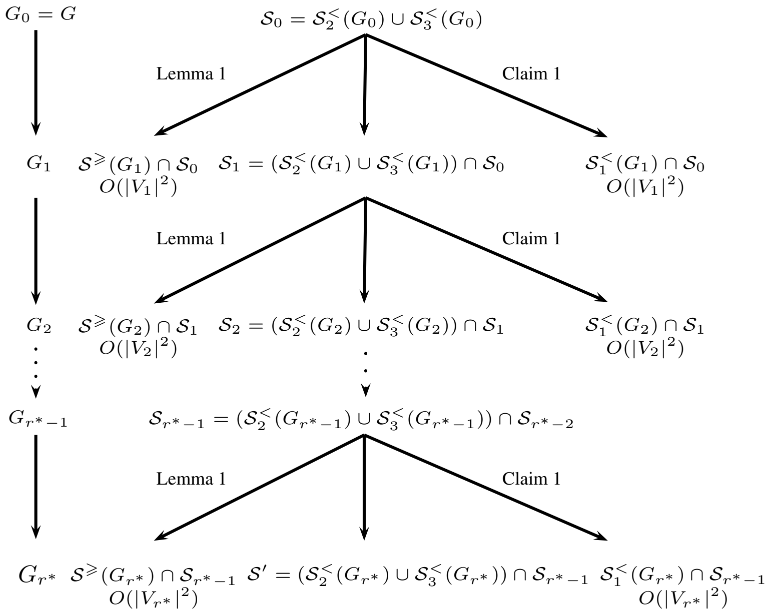

Fig. 2. The behavior of Algorithm 1

An upper bound on the cardinality of the remaining set $\mathcal{S}_1 = (\mathcal{S}_2^{<}(G_1) \cup \mathcal{S}_3^{<}(G_1)) \cap \mathcal{S}_0$ will be computed recursively. The algorithm keeps contracting edges until either the residual graph only contains two nodes or the cost $c^1(e)$ of every non-loop edge e is not in $\left[\frac{c^1(E_r)}{|V_r|}, \frac{c^1(E_r)}{2}\right]$. Let r^* denote the number of iterations of Algorithm 1. The algorithm returns the cardinality of the set $\mathcal{S} = (\mathcal{S}_0 \cap (\mathcal{S}^{\geqslant}(G_1) \cup \mathcal{S}_1^{<}(G_1))) \cup \cdots \cup (\mathcal{S}_{r^*-1} \cap (\mathcal{S}^{\geqslant}(G_{r^*}) \cup \mathcal{S}_1^{<}(G_{r^*})))$. If the last graph G_{r^*} contains more than two nodes, then additional work needs to be done in order to give an upper bound on the cardinality of the set $\mathcal{S}' = (\mathcal{S}_2^{<}(G_{r^*}) \cup \mathcal{S}_3^{<}(G_{r^*})) \cap \mathcal{S}_{r^*-1}$.

The following claim establishes a relation between $\mathcal{S}_0$ and intermediate sets $\mathcal{S}(G_r)$, $\mathcal{S}_1^{<}(G_r)$, $\mathcal{S}_2^{<}(G_r)$ and $\mathcal{S}_3^{<}(G_r)$ generated by the algorithm (proof omitted).

Claim 1. For iteration r, we have

$$\mathcal{S}_0 \subseteq (\mathcal{S}_{r-1} \cap (\mathcal{S}_2^{<}(G_r) \cup \mathcal{S}_3^{<}(G_r))) \cup (\cup_{l=1}^{r} (\mathcal{S}_{l-1} \cap (\mathcal{S}^{\geqslant}(G_l) \cup \mathcal{S}_1^{<}(G_l)))), \quad (2)$$

and

$$|\mathcal{S}_0| \leqslant |\mathcal{S}_{r-1} \cap (\mathcal{S}_2^{<}(G_r) \cup \mathcal{S}_3^{<}(G_r))| + O(r|V|^2). \quad (3)$$

Algorithm 1.

let $E_0 \leftarrow E$, $V_0 \leftarrow V$, $G_0 \leftarrow G$, $r \leftarrow 0$, $\mathcal{S} \leftarrow \emptyset$, $\mathcal{S}_0 \leftarrow \mathcal{S}_2^{<}(G_0) \cup \mathcal{S}_3^{<}(G_0)$, $test \leftarrow$ true
repeat
  if there exists a non-loop edge $e = (u, v) \in E_r$ such that $\frac{c^1(E_r)}{|V_r|} \leqslant c^1(e) \leqslant \frac{c^1(E_r)}{2}$ **then**
    contract e
    $r \leftarrow r + 1$
    denote by $G_r = (V_r, E_r)$ the resulting graph such that $V_r = (V_{r-1} \setminus \{u, v\}) \cup \{w\}$ where w is the node obtained by merging u and v, and $E_r = E_{r-1} \setminus \{e\}$ (the loops are kept)
    apply oracle $\mathbf{O}$ for G_r and compute $\mathcal{S}(G_r)$
    denote by $\mathcal{S}_r = (\mathcal{S}_2^{<}(G_r) \cup \mathcal{S}_3^{<}(G_r)) \cap \mathcal{S}_{r-1}$
    $\mathcal{S} \leftarrow \mathcal{S} \bigcup ((\mathcal{S}^{\geqslant}(G_r) \cup \mathcal{S}_1^{<}(G_r)) \cap \mathcal{S}_{r-1})$
  else
    $test \leftarrow$ false
  end if
until $|V_r| = 2$ or $test =$ false
Output: $|\mathcal{S}|$

Observe that, in general, the bound given in (3) is not tight since sets $\mathcal{S}_{r-1} \cap (\mathcal{S}^{\geqslant}(G_r) \cup \mathcal{S}_1^{<}(G_r))$ might not be disjoint.

At the end of Algorithm 1, three cases may happen. First, if G_{r^*} contains only two nodes, then by Claim 1, we have $|\mathcal{S}_0| \leqslant |\mathcal{S}| + 1 \leqslant O(|V|^3)$. Next, if $c^1(e) < \frac{c^1(E_{r^*})}{|V_{r^*}|}$ for all non-loop edge $e \in E_{r^*}$, then the problem reduces to computing an upper bound for $|\mathcal{S}'|$. Here we can show that $|\mathcal{S}'| \leqslant \frac{|V_{r^*}|}{2}$. Finally, if both previous cases do not hold, then there exist a non-loop edge $\bar{e} \in E_{r^*}$ such that $c^1(\bar{e}) > \frac{c^1(E_{r^*})}{2}$ and $c^1(e) < \frac{c^1(E_{r^*})}{|V_{r^*}|}$ for all non-loop edges $e \in E_{r^*} \setminus \{\bar{e}\}$. This case can be handled in a similar way as the previous one. Therefore, we only focus in the rest of the proof on the second case.

Algorithm 2. (G_{r^*})

$E'_0 \leftarrow E_{r^*}$, $V'_0 \leftarrow V_{r^*}$, $G'_0 \leftarrow G_{r^*}$, $r \leftarrow 0$,
while $|V'_r| > 2$ **do**
  choose an edge $e = (u, v) \in E'_r$ with probability $\frac{c^1(e)}{c^1(E'_r)}$
  if e is not a loop **then**
    contract e and remove it from the residual graph
    $r \leftarrow r + 1$
    denote by $G'_r = (V'_r, E'_r)$ the graph such that $V'_r = (V'_{r-1} \setminus \{u, v\}) \cup \{w\}$ where w is a node obtained from merging u and v, and $E'_r = E'_{r-1} \setminus \{e\}$ (the loops are kept)
  end if
end while
return the unique cut

For this purpose, it is sufficient to show that $|\mathcal{S}_2^<(G_{r^*}) \cup \mathcal{S}_3^<(G_{r^*})| \leqslant \frac{|V_{r^*}|}{2}$. This is done using Algorithm 2 based on probabilistic arguments similar to Karger's algorithm [6].

Algorithm 2 has as input graph G_{r^*} provided by Algorithm 1. In each iteration r, the current graph is denoted by $G'_r = (V'_r, E'_r)$ and the algorithm randomly chooses an edge e with probability $\frac{c^1(e)}{c^1(E'_r)}$. If e is a non-loop edge, then it is contracted. This process continues until the last graph only contains two nodes. Then the algorithm returns the unique cut in this graph. By hypothesis, $\mu c^1(X) + (1-\mu)c^2(X) > 0$ for every $X \in \mathcal{S}(G)$ and $\mu \in I(G, X)$. Thus $c^1(E'_r) > 0$ for $r \leqslant V_{r^*-2}$ and the probability of edge selection is always defined. As in Algorithm 1, loops resulting from contractions are not removed from residual graphs. However, by contrast to [6,7], Algorithm 2 has a pseudo-polynomial expected running time.

Claim 2. Algorithm 2 has a pseudo-polynomial expected running time.

Proof. Given an integer $r \in \{1, \ldots, |V_{r^*}| - 2\}$, let N_r denote a random variable defining the number of iterations of the while loop separating two consecutive contraction operations, say the r^{th} and $r + 1^{\text{st}}$, and let N be a random variable defining the total number of iterations of the algorithm. We have $E(N) = |V_{r^*}| - 2 + \sum_{l=1}^{|V_{r^*}|-2} E(N_l)$. Let $\bar{E}_r$ denote the set of loops in the current graph G'_r. N_r is a geometric random variable with parameter $p_r = \frac{\sum_{e \in E'_r \setminus \bar{E}_r} c^1(e)}{c^1(E'_r)}$. Thus $E(N_r) = \frac{1}{p_r} \leqslant c^1(E'_r) \leqslant c^1(E)$ and $E(N) \leqslant (|V_{r^*}| - 2)(1 + c^1(E))$. ■

Claim 3. Algorithm 2 returns any solution in $\mathcal{S}_2^<(G_{r^*}) \cup \mathcal{S}_3^<(G_{r^*})$ with probability at least $\frac{2}{|V_{r^*}|}$.

Proof. Consider any solution $X \in \mathcal{S}_2^<(G_{r^*}) \cup \mathcal{S}_3^<(G_{r^*})$. Algorithm 2 returns X only if none of its edges has been contracted. Therefore, no error occurs if a loop is selected. In the rest of the proof, we only focus on iterations of the while loop where a non-loop edge is chosen. Suppose that r edges not in X have been contracted through the algorithm. Since loops are not removed, $|E'_r| = |E'_{r-1}| - 1$. The probability that an edge from X is selected in the $r + 1^{th}$ contraction operation is $\frac{c^1(X)}{c^1(E'_r)} \leqslant \frac{c^1(E'_0)}{|V'_0| c^1(E'_r)}$. Since $c^1(e) < \frac{c^1(E'_0)}{|V'_0|}$ for every non-loop edge $e \in E'_0$, it follows that $c^1(E'_r) \geqslant c^1(E'_0)(1 - \frac{r}{|V'_0|})$. Therefore, the probability for error is at most $\frac{1}{|V'_0| - r}$, and the probability that no edge of X is chosen after the $r + 1^{th}$ contraction operation is at least $\frac{|V'_0| - r - 1}{|V'_0| - r}$. Hence, the probability that no edge from X is never chosen (after $|V'_0| - 2$ contraction operations) is at least $\frac{|V'_0|-1}{|V'_0|} \cdot \frac{|V'_0|-2}{|V'_0|-1} \cdot \frac{|V'_0|-3}{|V'_0|-2} \cdots \frac{2}{3} = \frac{2}{|V'_0|} = \frac{2}{|V_{r^*}|}$. ■

Since the probability that a cut in $\mathcal{S}_2^<(G_{r^*}) \cup \mathcal{S}_3^<(G_{r^*})$ survives all the contraction operations is at least $\frac{2}{|V_{r^*}|}$, and that no two cuts can survive simultaneously, it follows that $|\mathcal{S}_2^<(G_{r^*}) \cup \mathcal{S}_3^<(G_{r^*})| \leqslant \frac{|V_{r^*}|}{2}$. Therefore, $|\mathcal{S}_0| \leqslant O(|V|^3)$.

Using a similar argument, we can also show that $|\mathcal{S}_1^<(G)|$ is $O(|V|^3)$, and the proof is complete. □

As a consequence of Theorem 1, we will show that the number of non-dominated points is also strongly polynomial.

Corollary 1 *The number of SND and UND points of the global minimum cut problem are $O(|V|^5)$ and $O(|V|^7)$, respectively.*

Proof. As it is proved in [6,8], the number of optimal solutions of the global minimum cut problem is $O(|V|^2)$. Thus, the bound on the number of SND points follows from combining this result and Theorem 1.

Now consider two SND points X_1, X_2. Suppose that they are optimal for some $\mu = \mu_1$. Let X_3 be a UND point dominated by a convex combination of X_1 and X_2. W.l.o.g., suppose that $c^1(X_1) < c^1(X_2)$ and $c^2(X_2) < c^2(X_1)$. We then have $c^1(X_3) < c^1(X_2)$ and $c^2(X_3) < c^2(X_1)$. Therefore,

$$\begin{aligned}\mu_1 c^1(X_3) + (1-\mu_1)c^2(X_3) &< \mu_1 c^1(X_2) + (1-\mu_1)c^2(X_1) \\ &\leqslant \mu_1 c^1(X_2) + (1-\mu_1)c^2(X_2) + \mu_1 c^1(X_1) + \\ &(1-\mu_1)c^2(X_1).\end{aligned}$$

Thus, X_3 is a 2-approximate solution for $\mu = \mu_1$. The bound on the number of UND points follows from [6] and Theorem 1. □

2.2 Efficient Algorithms for *f* and the Non-dominated Points Set

We are in the single-parameter case ($k = 2$; the full paper shows how to extend these algorithms for $k > 2$), and we want to compute f for all $\mu \in [0, 1]$. Our strongly polynomial algorithm is based on the Discrete Newton Algorithm [10,18]. For a fixed $\mu_1 \in [0, 1]$, using a cactus representation of minimum cuts [15] we compute optimal cuts $X^*_+(\mu_1)$ and $X^*_-(\mu_1)$ at μ_1 such that $X^*_+(\mu_1)$ ($X^*_-(\mu_1)$) is the optimal cut X whose line $\mu_1 c^1(X) + (1-\mu_1)c^2(X)$ has the largest (smallest) slope among all optimal cuts at μ_1. Let *Steepest*$^+$*(μ_1)* and *Steepest*$^-$*(μ_1)* be the lines associated with $X^*_+(\mu_1)$ and $X^*_-(\mu_1)$. Consider Algorithm 3.

Algorithm 3. Discrete Newton method

$L \leftarrow \{[0, 1]\}$, $\mathcal{B} \leftarrow \emptyset$
while $L \neq \emptyset$ **do**
  choose an interval $[\mu_1, \mu_2] \in L$ and compute *Steepest*$^-(\mu_1)$ and *Steepest*$^+(\mu_2)$
  compute $\mu_3 \in [\mu_1, \mu_2]$ corresponding to the intersection of *Steepest*$^-(\mu_1)$ and *Steepest*$^+(\mu_2)$
  if $\min_C\{\mu_3 c^1(C) + (1-\mu_3)c^2(C) : \emptyset \neq C \neq V\} = \mu_3 c^1(X^*_-(\mu_1)) + (1-\mu_3)c^2(X^*_-(\mu_1)) = \mu_3 c^1(X^*_+(\mu_2)) + (1-\mu_3)c^2(X^*_+(\mu_2))$ **then**
    $L \leftarrow L \setminus \{[\mu_1, \mu_2]\}$ and $\mathcal{B} \leftarrow \mathcal{B} \cup \{\mu_3\}$
  else
    $L \leftarrow L \setminus \{[\mu_1, \mu_2]\} \cup \{[\mu_1, \mu_3], [\mu_3, \mu_2]\}$
  end if
end while
Return $\mathcal{B}$

Algorithm 3 manages a list L of unexplored intervals containing at least one breakpoint and a list $\mathcal{B}$ of breakpoints. In each iteration the algorithm chooses an interval $[\mu_1, \mu_2] \in L$ and computes μ_3 as the intersection of the lines *Steepest*$^-(\mu_1)$ and *Steepest*$^+(\mu_2)$. It is clear that these lines are part of the function f. If the condition in Step 5 holds, then μ_3 is the unique breakpoint in $[\mu_1, \mu_2]$ and so we can fathom $[\mu_1, \mu_2]$ and add μ_3 to $\mathcal{B}$. Otherwise, $[\mu_1, \mu_3]$ and $[\mu_3, \mu_2]$, each contains at least one breakpoint, and $[\mu_1, \mu_2]$ is replaced by them. Using this, we obtain the following.

Proposition 1 *Algorithm 3 has $O(|E||V|^4 + |V|^5 \log |V|)$ running time.*

Proof. For any facet of function f, Algorithm 3 computes a global minimum cut and finds a cactus representation for at most three values: the two extremities and an intermediate value. Therefore, the total number of iterations is at most twice the number of facets. A cactus representation can be obtained in time $O(|E||V| + |V|^2 \log |V|)$ [15] and a global minimum cut can be computed in the same time complexity [13]. Therefore, the time complexity follows by Theorem 1. □

Since the time required to compute UND points dominates that of SND points, we have:

Proposition 2 *The time required to compute all the non-dominated points of the global minimum cut problem is $O(|E||V|^7)$.*

Proof. Computing all the 2-approximate solutions can be performed in time $O(|E||V|^4)$ [16]. The result follows by combining this and Theorem 1. □

3 Extensions

3.1 Parametric Complexity with More Than Two Criteria and Arbitrary Cost Functions

First, suppose we are given a graph $G = (V, E)$, and two cost functions $c^1, c^2 : E \to \mathbb{R}$. Here we allow some cost components to be negative. For $\mu \in \mathbb{R}$ define the parametric edge costs $f_\mu(e) = \mu c_e^1 + c_e^2$. We suppose that there exists an interval $[\alpha, \beta]$ where all $f_\mu(e)$ are positive. Here Lemma 1 and Step 3 of Algorithm 2 are no longer applicable and thus, Theorem 1 does hold in this case. Using ideas in [11] and avoiding unnecessary subdivisions of the interval $[\alpha, \beta]$, we obtain a strongly polynomial bound. The next result is also non-constructive (proof omitted).

Theorem 2 *Assume that the parametric edge costs $f_\mu(e)$ are positive for all $\mu \in [\alpha, \beta]$. The parametric complexity of the global min cut problem is $O(|E|^2|V|^2 \log |V|)$.*

It is natural to wonder if Theorem 3 could be extended to a constant number of parameters at least equal to two. More precisely, we are given $k \geqslant 3$ cost functions $c^1, \ldots, c^k : E \to \mathbb{R}$. For $\mu = (\mu_1, \mu_2) \in \mathbb{R}^{k-1}$, define the parametric edge costs $f_\mu(e) = \sum_{i=1}^{k-1} \mu_i c_e^i + c_e^k$ for $e \in E$. Assume that there exists an hyper-rectangle $I = [\alpha_1, \beta_1] \times \cdots \times [\alpha_{k-1}, \beta_{k-1}]$ such that $f_\mu(e)$ are positive for all $\mu \in I$ and $e \in E$. The next result shows that the parametric complexity is again strongly polynomial in this case (proof omitted).

Theorem 3 *Assume that the parametric edge costs $f_\mu(e)$ are positive for all $\mu \in I$. Then the parametric complexity of the global min cut problem is $O(|E|^k |V|^2 \log^{k-1} |V|)$.*

3.2 Hypergraphs

We consider finite hypergraphs $H = (V, E)$, where V is a finite set of *nodes* and each *edge* $e \in E$ is a subset of V. Hypergraph H is *rank*-ρ if every edge in H has cardinality at most ρ (e.g., a graph is a rank-ρ hypergraph for every $\rho \geq 2$).

A *cut* C in $H = (V, E)$ is any nontrivial node subset, i.e., satisfying $\emptyset \neq C \subset V$. Let $\Delta(C) = \{e \in E : e \cap C \neq \emptyset \neq e \setminus C\}$ denote the set of edges crossed by the cut C. Given nonnegative edge costs $c(e) \geq 0$ $(e \in E)$, let $c(F) = \sum_{e \in F} c(e)$ be the total cost of all edges in subset $F \subseteq E$, and let $c(C) \equiv c(\Delta(C))$ denote the total cost of all the edges crossed by the cut $C \subset V$. Further define $c(H) = \min_C \{c(C) : \emptyset \neq C \subset V\}$ as the minimum cost of a cut in H. There exist polynomial time algorithms for finding a minimum cost cut in a hypergraph, see [9,12,17].

The technique used to derive an upper bound on the parametric complexity can be generalized to hypergraphs. In order to extend Lemma 1 to hypergraphs, we now bound the number of approximate cuts in hypergraphs.

Theorem 4 *For any fixed integer $\rho \geq 2$ and scalar $\alpha \geq 1$, and rank-ρ hypergraph $H = (V, E)$ with nonnegative edge costs c and positive minimum cut cost, the number of cuts C with cost $c(C) \leq \alpha\, c(H)$ is $O\left(|V|^{B(\rho,\alpha)}\right)$ where*

$$B(\rho, \alpha) = \begin{cases} 2 & \text{if } \rho \leq 3 \text{ and } \alpha < \frac{3}{2}; \\ 2\alpha & \text{if } \rho \leq 3 \text{ and } \alpha \geq \frac{3}{2}; \\ \frac{\rho}{2} + \frac{2}{3} & \text{if } \rho \geq 4 \text{ and } \alpha < \frac{3}{2}; \\ \left(\frac{\rho}{2} + \frac{2}{3}\right)\alpha & \text{otherwise, i.e., if } \rho \geq 4 \text{ and } \alpha \geq \frac{3}{2}. \end{cases}$$

Proof. W.l.o.g., assume that every edge $e \in E$ has cardinality $|e| \geq 2$ (since edges e with $|e| \leq 1$ are not crossed by any cut). We prove the result by approximating the minimum cut problem in hypergraph $H = (V, E)$ with edge costs c by the minimum cut problem in the complete graph $K(V) = (V, E_{K(V)})$ with edge costs c', whereby each edge $e \in E$ is replaced by a clique on the nodes in e, where each edge in the clique has cost $c_e/(|e| - 1)$, i.e., by letting

$$c'_{i,j} = \sum_{e \in E : \{i,j\} \subseteq e} \frac{c_e}{|e| - 1} \quad \text{for all } \{i, j\} \in E_{K(V)}.$$

In particular, every cardinality-2 edge $e = \{i, j\} \in E$ contributes its full cost c_e to $c'_{i,j}$, and thus to the cost $c'(C)$ of every cut C in $K(V)$ that crosses it. Note also that every cardinality-3 edge $e \in E$ that is crossed by cut C has two of its nodes on one side of the cut and the other node on the other side, and thus also contributes its exact cost $2(c_e/2) = c_e$ to $c'(C)$. Therefore (as it is well known, e.g., Ihler et al. [5]), when $\rho \leq 3$ this transformation is exact, i.e., $c'(C) = c(C)$ for every cut C. The first two cases in the definition of $B(\rho, \alpha)$ then follow from Henzinger and Williamson [4] and Karger and Stein [8], respectively.

Now assume that $\rho \geq 4$. A cut C that crosses an edge $e \in E$ with cardinality $|e| \geq 4$ crosses at least $|e| - 1$ edges in the clique $K(e)$ (when exactly one node of e is on one side of the cut), and at most $|e|^2/4$ such edges (when half the nodes of e are on either side). Thus every edge $e \in E$ crossed by C contributes at least c_e and at most $(|e|^2/4)\frac{c_e}{|e|-1} \leq \frac{\rho^2/4}{\rho-1}\, c_e$ to the cost $c'(C)$. Therefore,

$$c(C) \leq c'(C) \leq \frac{\rho^2/4}{\rho - 1}\, c(C) = \left(\frac{\rho}{4} + \frac{1}{4}\left(1 + \frac{1}{\rho - 1}\right)\right) c(C) \leq \beta(\rho)\, c(C),$$

where $\beta(\rho) = \frac{\rho}{4} + \frac{1}{3}$ and the last inequality follows from $\rho \geq 4$. Let C' denote a minimum cut for $(K(V), c')$. If C is an α-optimal cut for (H, c), i.e., $c(C) \leq \alpha\, c(H)$, we have

$$c'(C) \leq \beta(\rho)\, c(C) \leq \beta(\rho)\, \alpha\, c(H) \leq \beta(\rho)\, \alpha\, c(C') \leq \beta(\rho)\, \alpha\, c'(C'),$$

implying that C is a $\big(\beta(\rho)\, \alpha\big)$-optimal cut for $(K(V), c')$. Then the last two cases in the definition of $B(\rho, \alpha)$ again follow from Henzinger and Williamson [4] and Karger and Stein [8], respectively. □

Theorem 5 *For any fixed scalar $\rho \geq 2$, let $H = (V, E)$ be a rank-ρ hypergraph with nonnegative edge costs c and $c^1, c^2 : E \to \mathbb{R}_+$ two nonnegative cost functions defined on its edges. Assume that the edge costs $f_\mu(e) = \mu c^1_e + (1 - \mu) c^2_e$, for all $e \in E$, are functions of a parameter $0 \leqslant \mu \leqslant 1$, and $\mu c^1(X) + (1 - \mu) c^2(X) > 0$ for any $X \in \mathcal{S}(G)$ and $\mu \in I(G, X)$. The parametric complexity of global minimum cut is $O\big(|V|^{B(\rho, \frac{3}{2})+1}\big)$.*

Proof. The proof is an adaptation of that of Theorem 1 to hypergraphs. By Theorem 4 and Lemma 1, we have $\mathcal{S}^{\geqslant}(H) \leqslant O\big(|V|^{B(\rho, \frac{3}{2})}\big)$. Now one can extend Algorithm 1 to hypergraphs, by contracting hyperedges instead of edges, and obtain that (2) holds in this case and (3) extends to

$$|\mathcal{S}_0| \leqslant |\mathcal{S}_{r-1} \cap (\mathcal{S}^{<}_2(G_r) \cup \mathcal{S}^{<}_3(G_r))| + O(r|V|^{B(\rho, \frac{3}{2})}). \tag{4}$$

Similarly, still three cases have to be considered. If G_{r^*} contains only two nodes, then by (4) we have $|\mathcal{S}_0| \leqslant O(|V|^{B(\rho, \frac{3}{2})+1})$. If $c^1(e) < \frac{c^1(E_{r^*})}{|V_{r^*}|}$ for all $e \in E_{r^*}$, then the problem reduces to computing an upper bound for $|\mathcal{S}'|$. Algorithm 2 and Claims 2-3 apply in this case and yield $|\mathcal{S}'| \leqslant \frac{|V_{r^*}|}{2}$. Therefore by (4), we have $|\mathcal{S}_0| \leqslant O(|V|^{B(\rho, \frac{3}{2})+1})$. Finally, the case where there exists a non-loop hyperedge $\bar{e} \in E_{r^*}$ such that $c^1(\bar{e}) > \frac{c^1(E_{r^*})}{2}$ (and $c^1(e) < \frac{c^1(E_{r^*})}{|V_{r^*}|}$ for all non-loop hyperedge $e \in E_{r^*} \setminus \{\bar{e}\}$) can be handled in a similar way as the previous one. Therefore, the result follows. □

References

1. Armon, A., Zwick, U.: Multicriteria global minimum cuts. Algorithmica 46(1), 15–26 (2006)
2. Carstensen, P.: Complexity of some parametric integer and network programming problems. Mathematical Programming 26, 64–75 (1983)
3. Ehrgott, M.: Multicriteria Optimization. Springer (2005)
4. Henzinger, M., Williamson, D.P.: On the number of small cuts in a graph. Information Processing Letters 59(1), 41–44 (1996)
5. Ihler, E., Wagner, D., Wagner, F.: Modeling hypergraphs by graphs with the same mincut properties. Information Processing Letters 45(4), 171–175 (1993)
6. Karger, D.R.: Global min-cuts in RNC, and other ramifications of a simple mincut algorithm. In: Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 21–30 (1993)
7. Karger, D.R.: Minimum cuts in near-linear time. Journal of the ACM 47(1), 46–76 (2000)
8. Karger, D.R., Stein, C.: A new approach to the minimum cut problem. Journal of the ACM 43(4), 601–640 (1996)
9. Klimmek, R., Wagner, F.: A simple hypergraph min cut algorithm. Freie Univ., Fachbereich Mathematik (1996)
10. McCormick, S.T., Ervolina, T.R.: Computing maximum mean cuts. Discrete Applied Math 52, 53–70 (1994)
11. Mulmuley, K.: Lower bounds in a parallel model without bit operations. SIAM Journal on Computing 28(4), 1460–1509 (1999)
12. Mak, W.K., Wong, D.F.: A fast hypergraph min-cut algorithm for circuit partitioning. Integration, the VLSI Journal 30(1), 1–11 (2000)
13. Nagamochi, H., Ibaraki, T.: Computing edge-connectivity in multigraphs and capacitated graphs. SIAM Journal on Discrete Mathematics 5(1), 54–66 (1992)
14. Nagamochi, H., Ibaraki, T.: Algorithmic aspects of graph connectivity. Cambridge University Press (2008)
15. Nagamochi, H., Nakamura, S., Ishii, T.: Constructing a cactus for minimum cuts of a graph in $O(mn + n^2 \log n)$ time and $O(m)$ space. Inst. Electron. Inform. Comm. Eng. Trans. Inform. Systems, 179–185 (2003)
16. Nagamochi, H., Nishimura, K., Ibaraki, T.: Computing all small cuts in undirected networks. SIAM Journal of Discrete Mathematics 10, 469–481 (1997)
17. Queyranne, M.: Minimizing symmetric submodular functions. Mathematical Programming 82(1-2), 3–12 (1998)
18. Radzik, T.: Newton's Method for Fractional Combinatorial Optimization. In: Proceedings of IEEE Annual Symp. of Foundations of Computer Science, pp. 659–669 (1992)
19. Stoer, M., Wagner, F.: A simple min-cut algorithm. Journal of the ACM 44(4), 585–591 (1997)