Algorithmica (2006) 46: 15–26
DOI: 10.1007/s00453-006-0068-x

Algorithmica

Multicriteria Global Minimum Cuts[1]

Amitai Armon[2] and Uri Zwick[2]

Abstract. We consider two multicriteria versions of the global minimum cut problem in undirected graphs. In the k-criteria setting, each edge of the input graph has k non-negative costs associated with it. These costs are measured in separate, non-interchangeable, units. In the AND-version of the problem, purchasing an edge requires the payment of *all* the k costs associated with it. In the OR-version, an edge can be purchased by paying *any one* of the k costs associated with it. Given k bounds $b_1, b_2, \ldots, b_k$, the basic multicriteria decision problem is whether there exists a cut C of the graph that can be purchased using a budget of b_i units of the ith criterion, for $1 \le i \le k$.

We show that the AND-version of the multicriteria global minimum cut problem is polynomial for any fixed number k of criteria. The OR-version of the problem, on the other hand, is NP-hard even for $k = 2$, but can be solved in pseudo-polynomial time for any fixed number k of criteria. It also admits an FPTAS. Further extensions, some applications, and multicriteria versions of two other optimization problems are also discussed.

Key Words. Multicriteria optimization, Minimum cut, Graph algorithms.

1. Introduction. We consider two multicriteria versions of the *global* minimum cut problem in undirected graphs. Let $G = (V, E)$ be an undirected graph, and let $w_1, \ldots, w_k$: $E \to \mathbb{R}^+$ be k non-negative cost (or weight) functions defined on its edges. A *cut* C of G is a subset $C \subseteq V$ such that $C \ne \emptyset$ and $C \ne V$. The edges cut by this cut are $E(C) = \{(u, v) \in E \mid u \in C, v \notin C\}$. (As the graph is undirected, C and $V - C$ define the same cut.) In the AND-version of the k-criteria problem, the ith weight (or cost) of the cut is

$$i\text{th cost in the AND-version:} \qquad w_i(C) = \sum_{e \in E(C)} w_i(e), \qquad 1 \le i \le k.$$

In the OR-version of the problem we pay only one of the costs associated with each edge $e \in E(C)$ of the cut. More specifically, we choose a function α: $E(C) \to \{1, 2, \ldots, k\}$ which specifies which cost is paid for each edge of the cut. The ith cost of the cut C, with respect to the choice function α, is then

$$i\text{th cost in the OR-version:} \qquad w_i(C, \alpha) = \sum_{e \in E(C) \wedge \alpha(e) = i} w_i(e), \qquad 1 \le i \le k.$$

For the AND-version, the basic multicriteria global minimum cut decision problem asks, given k cost bounds $b_1, b_2, \ldots, b_k$, is there a cut C such that $w_i(C) \le b_i$, for $1 \le i \le k$?

---

[1] A preliminary version of this paper appeared in the proceedings of ISAAC '04.
[2] School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. {armon,zwick}@cs.tau.ac.il.

Received September 4, 2004; revised September 16, 2005. Communicated by R. Fleischer.
Online publication June 19, 2006.

In the optimization problem we are given $k-1$ bounds $b_1, b_2, \ldots, b_{k-1}$ and asked to find a cut C for which $w_k(C)$ is minimized, subject to the constraints $w_i(C) \le b_i$, for $1 \le i \le k-1$. The *min-max* version of this problem asks for a cut C for which $\max_{i=1}^{k} w_i(C)$ is minimized, i.e., a cut whose largest cost is as small as possible. The *Pareto set* $P(G, w_1, \ldots, w_k) \subseteq \mathbb{R}^k$ of an instance $\langle G, w_1, \ldots, w_k\rangle$ is the set of cost vectors of cuts that are not dominated by the cost vector of any other cut. It follows, therefore, that if $(c'_1, c'_2, \ldots, c'_k)$ is the cost vector of a cut C' of the graph, then there exists a vector $(c_1, c_2, \ldots, c_k) \in P(G, w_1, \ldots, w_k)$ such that $c_i \le c'_i$ for $1 \le i \le k$. Corresponding definitions can be made for the OR-version of the problem, where α should then also be chosen.

Multicriteria optimization is an active field of research (see, e.g., the books of Climacao [3] and Ehrgott [4]). (Most research is focused, in our terminology, on AND-versions of various optimization problems. All results cited below refer to the AND-versions of the problems, unless stated otherwise.) Papadimitriou and Yannakakis [18] investigated the complexity of several multicriteria optimization problems. In particular, they considered the multicriteria s-t minimum cut problem, in which the cut must separate two specified vertices, s and t. They proved that this problem is strongly NP-complete, even for just two criteria.

We show here that (the AND-version of) the multicriteria global minimum cut decision problem can be solved in polynomial time for any *fixed* number of criteria, making it strictly easier than its s-t variant. The running time of our algorithm is $O(mn^{2k})$, where $m = |E|$ is the number of edges in the graph, $n = |V|$ is the number of vertices, and k is the number of criteria. This easily implies tractability of the optimization problem and also yields a pseudo-polynomial algorithm for constructing the Pareto set. The problem, however, becomes strongly NP-hard when the number of criteria is not fixed. We also show that the directed version of the problem is strongly NP-hard even for just two criteria.

The single-criterion minimum cut problem has been studied for more than four decades as a fundamental graph optimization problem (see, e.g., [6], [16], [14], [20], [12], and [13]). Minimum cuts are used in solving a large variety of problems, including VLSI design, network design and reliability, clustering, and more (see [13] and references therein). The best known deterministic algorithms for this problem run in $O(mn + n^2 \log n)$ time [16], [20]. The best known randomized algorithm runs in $O(m \log^3 n)$ time [13]. (As can be seen, there is a huge gap between the complexities of the deterministic and randomized algorithms!) These algorithms are faster than the best known algorithms for the s-t minimum cut problem which are based on network flow.

The polynomial time algorithm for the multicriteria problem relies on the fact that the standard single criterion global minimum cut problem has only a polynomial number of almost optimal solutions. More specifically, Karger and Stein [14] showed that for every $\alpha \ge 1$, not necessarily integral, the number of α-approximate solutions is only $O(n^{2\alpha})$. Karger [13] improved this bound to $O(n^{\lfloor 2\alpha \rfloor})$. Nagamochi et al. [17] gave a deterministic $O(m^2 n + mn^{2\alpha})$ time algorithm for finding all the α-approximate cuts. Our algorithms for the multicriteria problem use their algorithm.

Apart from the theoretical interest in minimum cuts in the multicriteria setting, there are some applications in which this problem is of interest. (The multicriteria global minimum cut problem is of interest in almost any application of the single criterion global minimum cut problem.) A multicriteria minimum balanced-partition is required, for example, in the situations described in [19].

A special case of the bicriteria global minimum cut problem, called the $\leq r$*-cardinality min-cut*, was considered by Bruglieri et al. [1], [2]. The input to this problem is an undirected graph $G = (V, E)$ with a single weight function $w\colon E \to \mathbb{R}^+$ defined on its edges. The goal is to find a cut of minimum cost that contains at most r edges. This is exactly the optimization version of the bicriteria minimum cut problem, where $w_1(e) = 1$, $w_2(e) = w(e)$, for every $e \in E$, and $w_1(C)$ must not exceed r. Bruglieri et al. [1], [2] ask whether this problem can be solved in polynomial time. We answer their question in the affirmative. We also obtain a polynomial time algorithm for finding a minimum cut which contains at most r vertices on the smallest side of the cut.

As mentioned, most research on multicriteria optimization focused, in our terminology, on AND-versions of various multicriteria optimization problems. We consider here also the OR-versions of the global minimum cut problem, the shortest path problem, and the minimum spanning tree problem.

OR-versions of multicriteria optimization problems may be seen as generalizations of the scheduling problem on *unrelated machines* (see [10], [15], and [11]). The input to such a scheduling problem is a set of n jobs that should be scheduled on m machines. The ith job has a cost vector $(c_{i1}, \ldots, c_{im})$ associated with it, where c_{ij} is the processing time of the ith job on the jth machine. The goal is to allocate the jobs to the machines so as to minimize the *makespan*, i.e., the completion time of the last job. (Jobs allocated to the same machine are processed sequentially.) This is precisely the OR-version of the min-max m-criteria minimum cut problem on a graph with two vertices and n parallel edges.

As another example where the OR-version of the multicriteria minimum-cut problem is of interest, consider a cyber-attacker wishing to disconnect a computer network, where there is more than one option for damaging each link. For example, assume that each link can either be disconnected by an electronic attack, which requires a certain amount of work hours (that may differ for different links), or by physically disconnecting it, e.g., by creating a strong electromagnetic field near the underground cable (the required power may again differ from link to link). Assuming an upper bound on the available power for electromagneric fields, what is the minimum electronic-attacks time which enables disconnecting the network?

It follows immediately from the simple reduction given above that the OR-version of the multicriteria global minimum cut problem is NP-hard even for just two criteria. We show, however, that the problem can be solved in pseudo-polynomial time for any fixed number of criteria. We also show that the problem can be solved in polynomial time when k, the number of criteria, is fixed and at least $k - 1$ of the weight functions assume only a fixed number of values. We also obtain some results on the complexity of the OR-versions of the shortest path and minimum spanning tree problems.

The rest of this paper is organized as follows. In the next section we consider the AND-version of the global minimum cut problem. In Section 3 we then consider the OR-version of the problem. In Section 4 we consider the OR-version of the multicriteria shortest path and minimum spanning tree problems. Finally, we conclude in Section 5 with some concluding remarks and open problems.

Algorithm Min-Max($G(V, E, w_1, \ldots, w_k)$):

1. Let $w'(e) = \sum_{i=1}^{k} w_i(e)$, for every $e \in E$.
2. Find all the k-approximate minimum cuts in G with respect to w'.
3. Among all the cuts C found in the previous step find the one for which $\max_{i=1}^{k} w_i(C)$ is minimized.

Fig. 1. A strongly polynomial time algorithm for the min-max version of the k-criteria global minimum cut problem.

2. Multicriteria Global Minimum Cut: The AND-Version. We first present a polynomial time algorithm for the min-max version of the multicriteria global minimum cut. The algorithm for solving the min-max version of the problem is then used to solve the decision and optimization problems.

2.1. *The Min-Max Problem.* An optimal min-max cut is a cut C for which $\max_{i=1}^{k} w_i(C)$ is minimized. We show that the simple algorithm given in Figure 1 solves the min-max version of the k-criteria global minimum cut problem in polynomial time, for every fixed k. A k-approximate cut in a graph G with respect to a single weight function w' is a cut whose weight is at most k times the weight of the minimum cut.

THEOREM 2.1. *Algorithm Min-Max solves the min-max version of the k-criteria global minimum cut problem. For any fixed k, it can be implemented to run, deterministically, in $O(mn^{2k})$ time.*

PROOF. We begin by proving the correctness of the algorithm. We show that if C is an optimal min-max cut, and D is any other cut in the graph, then $w'(C) \leq k \cdot w'(D)$, where $w'(e) = \sum_{i=1}^{k} w_i(e)$, for every $e \in E$. This follows as

$$w'(C) = \sum_{i=1}^{k} w_i(C) \leq k \cdot \max_{i=1}^{k} w_i(C) \leq k \cdot \max_{i=1}^{k} w_i(D) \leq k \cdot \sum_{i=1}^{k} w_i(D) = k \cdot w'(D).$$

The inequality $k \cdot \max_{i=1}^{k} w_i(C) \leq k \cdot \max_{i=1}^{k} w_i(D)$ follows from the assumption that C is an optimal min-max cut. In particular, if D is an optimal minimum cut with respect to the single weight function w', then $w'(C) \leq k \cdot w'(D)$, and it follows that C is a k-approximate cut of G with respect to w'. This proves the correctness of the algorithm.

We next consider the complexity of the algorithm. Karger and Stein [14] showed that every graph has at most $O(n^{2k})$ k-approximate cuts and gave a randomized algorithm for finding an *implicit* representation of them all in $\tilde{O}(n^{2k})$ time. A deterministic algorithm of Nagamochi et al. [17] explicitly finds all the k-approximte cuts in $O(mn^{2k})$ time. Choosing the best min-max cut among all the k-approximate cuts also takes only $O(mn^{2k})$ time. □

It is also easy to see that for any $1 < \alpha \leq k$, we can find an α-approximate solution to the min-max problem in $O(mn^{2k/\alpha})$ time, by checking all the k/α-approximate cuts in G'.

The randomized algorithm of Karger and Stein [14] extends to finding all the k-approximate minimum r*-cuts* in $\tilde{O}(n^{2k(r-1)})$ time. (An r*-cut* is a partition of the graph vertices into r sets, instead of 2). Thus, it is easy to see that the min-max multicriteria problem can also be solved for r*-cuts*, in $\tilde{O}(mn^{2k(r-1)})$ time using this randomized (Monte-Carlo) algorithm.

2.2. *The Decision Problem*. We next show that the algorithm for the min-max version of the k-criteria problem can be used to solve the decision version of the problem: Given k bounds $b_1, b_2, \ldots, b_k$, is there a cut C such that $w_i(C) \leq b_i$, for $1 \leq i \leq k$?

THEOREM 2.2. *For any fixed* k, *the decision version of the* k*-criteria global minimum cut problem can be solved*, *deterministically*, *in* $O(mn^{2k})$ *time*.

PROOF. The decision problem can be easily reduced to the min-max problem. Given k weight functions $w_1, \ldots, w_k$: $E \rightarrow R^+$ and k bounds $b_1, b_2, \ldots, b_k$, we simply produce *scaled* versions $w'_i(e) = w_i(e)/b_i$, for every $e \in E$ and $1 \leq i \leq k$, of the weight functions. Clearly the answer to the decision problem is 'yes' if and only if there is a cut C for which $\max_{i=1}^{k} w'_i(C) \leq 1$. □

2.3. *The Optimization Problem*. We next tackle the optimization problem: Given $k-1$ bounds $b_1, b_2, \ldots, b_{k-1}$, find a cut C for which $w_k(C)$ is minimized, subject to the constraints $w_i(C) \leq b_i$, for $1 \leq i \leq k-1$.

THEOREM 2.3. *For any fixed* k, *the optimization version of the* k*-criteria global minimum cut problem for graphs with integer edge weights can be solved*, *deterministically*, *in* $O(mn^{2k} \log M)$ *time*, *where* $M = \sum_{e \in E} w_k(e)$.

PROOF. If the kth weight function assumes only integral values, we can easily use binary search to solve the optimization problem. Given the $k-1$ bounds $b_1, b_2, \ldots, b_{k-1}$, we conduct a binary search for the minimal value b_k for which there is a cut C such that $w_i(C) \leq b_i$, for $1 \leq i \leq k$. As the minimal b_k is an integer in the range $[0, M]$, this requires the solution of only $O(\log M)$ decision problems. □

The algorithm given above is not completely satisfactory as it is not strongly polynomial and does not work with non-integral weights. These problems can be fixed, however, as we show below.

THEOREM 2.4. *For any fixed* k, *the optimization version of the* k*-criteria global minimum cut problem for graphs with arbitrary real edge weights can be solved*, *deterministically*, *in* $O(mn^{2k} \log n)$ *time*.

PROOF. We first isolate a small interval that contains the minimal value of b_k. Let $S = \{w_k(e) \mid e \in E\}$ be the set of values assumed by the kth weight function. The minimum b_k of the optimization problem lies in an interval $[s, ms]$, for some $s \in S$. (If C is the cut that attains the optimum, let s be the weight of the heaviest edge, with respect to w_k, in the cut.) Using a binary search on the values in S, we can find such an interval that contains the minimum. This requires the solution of only $O(\log m) = O(\log n)$ decision problems. Next, we conduct a binary search in the interval $[s, ms]$ until we narrow it down to an interval of the form $[s', (1 + 1/n)s']$ which is guaranteed to contain the right answer. This again requires the solution of only $O(\log(mn)) = O(\log n)$ decision problems.

Next, we run a modified version of the Min-Max algorithm given in Figure 1 on the following scaled versions of the weights: $w'_i(e) = w_i(e)/b_i$, for $1 \leq i \leq k - 1$, and $w'_k(e) = w_k(e)/s'$. It is easy to see that if C is an optimal solution of the optimization problem, then C is also a $(1 + 1/n)$-approximate solution of the min-max problem. This in turn implies, as in the proof of Theorem 2.1, that C is also a $k(1 + 1/n)$-approximate minimum cut with respect to the weight function $w'(e) = \sum_{i=1}^{k} w_i(e)$, for every $e \in E$. Instead of finding all the k-approximate minimum cuts with respect to w', as done by algorithm Min-Max, we find all the $k(1 + 1/n)$-approximate minimum cuts. Among all these cuts we find a cut C for which $w'_i(C) \leq 1$, i.e., $w_i(C) \leq b_i$, for $1 \leq i \leq k - 1$, and for which $w_k(C)$ is minimized. This cut is the optimal solution to the optimization problem.

We next analyze the complexity of the algorithm. The $O(\log n)$ decision problems can be solved in $O(mn^{2k} \log n)$ time. All the $k(1 + 1/n)$-approximate cuts can then be found in $O(mn^{2k(1+1/n)}) = O(mn^{2k})$ time using the algorithm of Nagamochi et al. [17]. (Note that $n^{1/n} = O(1)$.) Checking all these cuts also takes only $O(mn^{2k})$ time. This completes the proof of the theorem. □

2.4. *Two Applications*

THEOREM 2.5. *Let $G = (V, E)$ be an undirected graph and let $w\colon E \to \mathbb{R}^+$ be a weight function defined on its edges. Let $1 \leq r \leq m$. Then there is a deterministic $O(mn^4 \log n)$ time algorithm for finding a cut of minimum weight that contains at most r edges.*

PROOF. We simply let $w_1(e) = 1$ and $w_2(e) = w(e)$, for every $e \in E$, and solve the optimization problem with $b_1 = r$. □

As mentioned in the Introduction, this solves an open problem raised by Bruglieri et al. [1], [2]. We also have:

THEOREM 2.6. *Let $G = (V, E)$ be an undirected graph and let $w\colon E \to \mathbb{R}^+$ be a weight function defined on its edges. Let $1 \leq r \leq n$. Then there is a deterministic $O(n^6 \log n)$ time algorithm for finding a cut of minimum weight with at most r vertices on its smaller side.*

PROOF. We set up two weight functions over a complete graph on $n = |V|$ vertices: $w_1(u, v) = 1$, for every $u, v \in V$, and $w_2(u, v) = w(u, v)$, if $(u, v) \in E$, and $w(u, v) = 0$, otherwise. We then find a cut C that minimizes $w_2(C)$ subject to the constraint $w_1(C) \leq r(n - r)$. □

2.5. *The Pareto Set*. Suppose all weight functions are integral. Let $M_i = \sum_{e \in E} w_i(e)$, for $1 \leq i \leq k$. The Pareto set can be trivially found by invoking the basic decision algorithm $\prod_{i=1}^{k} M_i$ times, or the optimization algorithm $\prod_{i=1}^{k-1} M_i$ times. These naive algorithms are pseudo-polynomial for every fixed k.

Using very similar ideas we can also obtain an FPTAS for finding an *approximate* Pareto set, a notion defined by Papadimitriou and Yannakakis [18]. It is defined as a set of feasible k-tuples, such that for every solution there is a k-tuple in the set within a factor of $(1 - \varepsilon)$ in all coordinates. More formally, the set $P_\varepsilon(G, w_1, \ldots, w_k)$ is a set of cost vectors of cuts in the graph such that for every cut C there exists $(c_1, \ldots, c_k) \in P_\varepsilon(G, w_1, \ldots, w_k)$ such that $(1 - \varepsilon)c_i \leq w_i(C)$ for $1 \leq i \leq k$. It is easy to see that we can find this set in polynomial time, by invoking the basic algorithm for the decision problem only for powers of $(1 - \varepsilon)$, instead of checking all the possible values.

2.6. *Hardness Results*

THEOREM 2.7. *The multicriteria minimum cut problem with a non-fixed number of criteria is strongly NP-complete.*

PROOF. We use a reduction from the *bisection width* problem (see problem ND17 of [5]): Given an unweighted input graph $G = (V, E)$ on $n = 2r$ vertices and a bound b, is there a bisection of the graph that cuts at most b edges? We transform such an instance in the following way: Assume that $V = \{1, 2, \ldots, n\}$. We add two vertices, s and t, and add edges connecting them to each of the vertices in V. Let $G' = (V', E')$ be the resulting graph. Each edge of G' is now assigned $n + 3$ weights. For $1 \leq i \leq n$, we let $w_i(s, i) = w_i(t, i) = 1$, and $w_i(e) = 0$ for all other edges. We assign $w_{n+1}(e) = 1$ for the edges of the form (s, i), $i \in V$, and $w_{n+1}(e) = 0$, otherwise. Similarly, we assign $w_{n+2}(e) = 1$ for the edges of the form (t, i), $i \in V$, and $w_{n+2}(e) = 0$, otherwise. Finally, $w_{n+3}(e) = 1$ for $e \in E$, and $w_{n+3}(e) = 0$, otherwise. It is now easy to see that G has a bisection of width at most b if and only if G' has a cut C for which $w_i(C) \leq 1$, for $1 \leq i \leq n$, $w_{n+1}(C)$, $w_{n+2}(C) \leq r$, and $w_{n+3}(C) \leq b$. $\square$

It is also not difficult to show that the *directed* multicriteria global minimum cut problem is strongly NP-complete, even for two criteria. In this problem we are given a directed graph $G = (V, E)$ with weight functions $w_1, \ldots, w_k \colon E \to R^+$. A solution consists of a cut C, and of a labelling of the two vertex sets it separates by S and T. The weights of each cut are the sums of the weights of the cut edges directed from S to T. Each of the above mentioned variants for the undirected multicriteria minimum cut problem can be considered here as well: the decision, optimization, min-max, and Pareto-set problems. In the *directed* multicriteria *s*-*t* min-cut problem, two vertices, s and t, are specified with the input, and the solution must satisfy $s \in S$ and $t \in T$.

THEOREM 2.8. *The* directed *multicriteria global minimum cut problem is strongly NP-complete, even for just two criteria.*

PROOF. We show this by a reduction from *undirected* multicriteria s-t-min-cut, which is strongly NP-hard, even for $k = 2$ [18].

An instance of the *undirected* bicriteria s-t-min-cut decision problem can be reduced to an instance of the *directed* bicriteria s-t-min-cut problem simply by replacing each edge by two anti-parallel directed edges with the same weight. Recall that in a *directed* s-t-min-cut only edges directed from S to T contribute to the cut weights ($s \in S$ and $t \in T$), so having edges in the opposite direction does not influence the solution.

This instance can then be reduced to an instance of the directed bicriteria *global* min-cut decision problem. We simply connect each vertex to s with edges having weight $m \cdot M + 1$ in both criteria (where M is the maximal weight), and do the same from t to all the other vertices (if some of these edges already exist then we replace them). We assume that at least one input bound satisfies $b_i < m \cdot M + 1$, otherwise the answer is trivially "yes". So a solution to this problem will necessarily have s and t on different sides, $s \in S$ and $t \in T$. Therefore a solution to this problem also solves the original problem, and it has the same weights. Thus, the *directed* multicriteria global min-cut decision problem is strongly NP-hard, even for $k = 2$. □

3. Multicriteria Global Minimum Cut: The OR-Version

3.1. *Relation to Scheduling on Unrelated Machines*. As mentioned in the Introduction, there is a trivial reduction from the scheduling on unrelated machines problem to the OR-version of the min-max multicriteria global minimum cut problem. Known hardness results for the scheduling problem (see [15]) then imply the following:

THEOREM 3.1. *The OR-version of the min-max multicriteria global minimum cut problem is NP-hard even for just two criteria. The problem with a non-fixed number of criteria cannot be approximated to within a ratio better than* $3/2$, *unless* P = NP.

The scheduling problem on a fixed number of unrelated machines can however be solved in pseudo-polynomial time. Horowitz and Sahni [10] present a simple branch-and-bound pseudo-polynomial algorithm for that problem which runs in $O(m^2(kM)^{k-1})$ time, where m is the number of jobs, k is the number of machines, and M is the optimal makespan. This immediately implies:

THEOREM 3.2. *Let* $G = (V, E)$ *be an undirected graph with* k *integral weight functions* $w_1, \ldots, w_k \colon E \to \mathbb{N}$ *defined on its edges. Let* C *be a cut in* G. *Then a choice function* $\alpha \colon E(C) \to \{1, 2, \ldots, k\}$ *which minimizes* $\max_{i=1}^{k} w_i(C, \alpha)$ *can be found in pseudo-polynomial time*.

Jansen and Porkolab [11] obtained an FPTAS for the unrelated machines scheduling problem, which runs in $O(m(k/\varepsilon)^{O(k)})$ time (for any fixed number k of machines). It can be used instead of the exact algorithm of [10] when approximate solutions are acceptable.

3.2. *The Min-Max Version*. We show that the simple algorithm given in Figure 2, which is a variant of the algorithm given in Figure 1, solves the OR-version of the min-max problem in pseudo-polynomial time, for any fixed number of criteria.

Algorithm Min-Max-Or($G(V, E, w_1, \ldots, w_k)$):

1. Let $w'(e) = \min_{i=1}^{k} w_i(e)$, for every $e \in E$.
2. Find all the k-approximate minimum cuts in G with respect to w'.
3. For each of the cuts C found in the previous step, find the best choice function α: $E(C) \to \{1, 2, \ldots, k\}$.
4. Output the best cut and choice function found.

Fig. 2. A pseudo-polynomial time algorithm for the min-max version of the k-criteria global minimum cut problem.

THEOREM 3.3. *The OR-version of the min-max k-criteria global minimum cut problem with integer edge weights can be solved in $O(m^2 n^{2k} (kM)^{k-1})$ time, where M is the optimal min-max value.*

PROOF. We begin again with the correctness proof. Let C be an optimal min-max cut and let α be the corresponding optimal choice function. Let D be any other cut. We show that $w'(C) \le k \cdot w'(D)$, where $w'(e) = \min_{i=1}^{k} w_i(e)$, for every $e \in E$. To see that, we let β: $E(D) \to \{1, 2, \ldots, k\}$ be a choice function for which $\beta(e) = i$ if $w_i(e) \le w_j(e)$, for every $1 \le j \le k$. Then

$$w'(C) \;\le\; k \cdot \max_{i=1}^{k} w_i(C, \alpha) \;\le\; k \cdot \max_{i=1}^{k} w_i(D, \beta) \;\le\; k \cdot w'(D).$$

The second inequality follows as (C, α) is an optimal solution of the min-max problem.

We next consider the complexity of the algorithm. The k-approximate cuts with respect to w' can be found again in $O(mn^{2k})$ time using the algorithm of Nagamochi et al. [17]. For each one of the $O(n^{2k})$ approximate cuts produced, we find an optimal choice function using the algorithm of Horowitz and Sahni [10]. The total running time is then $O(mn^{2k} + n^{2k} \cdot m^2 (kM)^{k-1}) = O(m^2 n^{2k} (kM)^{k-1})$, where M is the value of the optimal solution. □

THEOREM 3.4. *The OR-version of the min-max k-criteria global minimum cut problem, with k fixed, admits an FPTAS.*

PROOF. The proof is identical to the proof of Theorem 3.3 with the exact algorithm of Horowitz and Sahni [10] replaced by the FPTAS of Jansen and Porkolab [11]. □

As in Section 2, we can use the algorithm for the min-max version of the problem to solve the decision and optimization versions of the problem. We omit the obvious details.

3.3. *A Case That Can Be Solved in Polynomial Time.* We now discuss a restriction of the min-max problem that can be solved in strongly polynomial time. For simplicity, we consider the bicriteria problem.

THEOREM 3.5. *Instances of the OR-version of the min-max bicriteria global minimum cut problem in which one of the weight functions assumes only r different values can be solved in* $O(m^{r+1}n^4)$ *time.*

PROOF. Assume, without loss of generality, that w_2 assumes only r different real values $a_1, a_2, \ldots, a_r$. Let $E_i = w_2^{-1}(a_i) = \{e \in E \mid w_2(e) = a_i\}$, for $1 \le i \le r$. Consider an optimal min-max cut C and an optimal choice function $\alpha\colon E(C) \to \{1, 2\}$ for it. It is easy to see that for every $1 \le i \le r$ there is a threshold t_i such that if $e \in E_i$, then $\alpha(e) = 1$ if and only if $w_1(e) \le t_i$. (Indeed, if there are two edges $e_1, e_2 \in E_i$ such that $w_1(e_1) < w_1(e_2)$, $\alpha(e_1) = 2$ and $\alpha(e_2) = 1$, then the choice function α' which reverses the choices of α on e_1 and e_2 is a better choice function. We assume here, for simplicity, that all weights are distinct.) As there are at most $m + 1$ essentially different thresholds for each set E_i, the total number of choice functions that should be considered is only $O(m^r)$. With a given choice function $\alpha\colon E \to \{1, 2\}$, the problem reduces to an AND-version of the problem with the weights $w_i'(e) = w_i(e)$, if $\alpha(e) = i$, and $w_i'(e) = 0$, otherwise, for $i = 1, 2$. As each such problem can be solved in $O(mn^4)$ time, the total running time of the resulting algorithm is $O(m^{r+1}n^4)$. □

4. OR-versions of Other Multicriteria Problems. In this section we consider the OR-versions of the bicriteria shortest path and minimum spanning tree problems. Our results can probably be extended to any fixed number of criteria.

4.1. *Shortest Paths*. The input to the problem is a directed graph $G = (V, E)$ with two weight functions $w_1, w_2\colon E \to \mathbb{R}^+$ defined on its edges, two vertices $s, t \in V$, and two bounds b_1, b_2. The question is whether there is a path P from s to t in the graph and a choice function $\alpha\colon P \to \{1, 2\}$ such that $w_1(\alpha^{-1}(1)) \le b_1$ and $w_2(\alpha^{-1}(2)) \le b_2$. The graph $G = (V, E)$ may, for example, represent the map of a city. Each edge $e \in E$ of the graph can be traversed either by bus or by subway. The weight $w_1(e)$ is the number of bus tokens needed for traversing the edge e by bus, while the weight $w_2(e)$ is the number of subway tokens needed to traverse e by subway. The question then is whether it is possible to get from s to t using given amounts of subway tokens and bus tokens.

It is easy to see, using a simple reduction from the scheduling on unrelated machines problem, that the OR-version of the bicriteria shortest path problem is NP-hard. We show, however, that it can be solved in pseudo-polynomial time. An FPTAS for the problem is easily obtained by scaling.

THEOREM 4.1. *The OR-version of the bicriteria shortest path decision problem with integer edge lengths can be solved in* $O(nmW \log(nW))$ *time, where* $W = \max_{e\in E} w_1(e)$.

PROOF. The OR-version of the problem can be easily reduced to the AND-version of the problem by replacing each edge e having a weight vector $(w_1(e), w_2(e))$ by two parallel edges e' and e'' having weight vectors $(w_1(e), 0)$ and $(0, w_2(e))$. The standard, AND-version, of the problem can be solved using an algorithm of Hansen [7] within the claimed time bound. □

4.2. *Minimum Spanning Trees*. Next we consider the OR-version of the bicriteria minimum spanning tree problem. The input is an undirected graph $G = (V, E)$, two weight functions $w_1, w_2\colon E \to \mathbb{R}$, and two bounds b_1 and b_2. The question is whether there exist a spanning tree T and a choice function $\alpha\colon T \to \{1, 2\}$ such that $w_1(\alpha^{-1}(1)) \le b_1$ and $w_2(\alpha^{-1}(2)) \le b_2$.

The OR-version of the bicriteria minimum spanning tree problem is again easily seen to be NP-hard, again using a simple reduction from the scheduling on unrelated machines problem. We provide a polynomial time algorithm for a special case of the problem, and a pseudo-polynomial time algorithm for the general case.

THEOREM 4.2. *The OR-version of the minimum spanning tree problem in which one of the weight functions is constant, i.e.*, $w_2(e) = c$, *for every* $e \in E$, *can be solved by solving a single standard minimum spanning tree problem.*

PROOF. We simply solve the standard minimum spanning tree problem with respect to the weight function w_1 and obtain a minimum spanning tree T . For the $\lfloor b_2/c \rfloor$ heaviest edges of T we choose to pay the w_2 cost, and for all the others we pay the w_1 cost. The correctness of this procedure follows from the well known fact that if the weights of the edges of T are $a_1 \le a_2 \le \cdots \le a_{n-1}$, and if T' is any other spanning tree of the graph G with edge weights $a'_1 \le a'_2 \le \cdots \le a'_{n-1}$, then $a_i \le a'_i$, for $1 \le i \le n-1$. $\square$

THEOREM 4.3. *The OR-version of the bicriteria minimum spanning tree decision problem with integer edge lengths can be solved in* $O(n^4 b_1 b_2 \log(b_1 b_2))$ *time.*

PROOF. The OR-version of the problem can be easily reduced to the AND-version of the problem by replacing each edge e having a weight vector $(w_1(e), w_2(e))$ by two parallel edges e' and e'' having weight vectors $(w_1(e), 0)$ and $(0, w_2(e))$. The standard, AND-version, of the problem can be solved using an algorithm of Hong et al. [9] within the claimed time bound. $\square$

We also note that an efficient PTAS for the AND-version of the bicriteria minimum spanning tree optimization problem was obtained recently by Hassin and Levin [8].

5. Concluding Remarks. We showed that the standard (i.e., the AND-version) multicriteria global minimum cut problem can be solved in polynomial time for any fixed number k of criteria. The running time of our algorithm, which is $O(mn^{2k})$, is fairly high, even for a small number of criteria. Improving this running time is an interesting open problem. We also considered the OR-version of the problem and showed that it is NP-hard even for just two criteria. It can be solved, however, in pseudo-polynomial time, and it also admits an FPTAS, for any fixed number of criteria. Finally, we considered the OR-versions of the bicriteria shortest path and minimum spanning tree problems, and showed that both of them are NP-hard but can be solved in pseudo-polynomial time. It will also be interesting to study OR-versions of other multicriteria optimization problems.

References

[1] M. Bruglieri, M. Ehrgott, and H. W. Hamacher. Some complexity results for k-cardinality minimum cut problems. Technical Report 69/2000, Wirtschaftsmathematik, University of Kaiserslautern, 2000.

[2] M. Bruglieri, F. Maffioli, and M. Ehrgott. Cardinality constrained minimum cut problems: Complexity and algorithms. *Discrete Applied Mathematics*, 137(3):311–341, 2004.

[3] J. Climacao. *Multicriteria Analysis*. Springer-Verlag, New York, 1997.

[4] M. Ehrgott. *Multicriteria Optimization*. Lecture Notes in Economics and Mathematical Systems, 491. Springer-Verlag, New York, 2000.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[6] R.E. Gomory and T. C. Hu. Multiterminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[7] P. Hansen. Bicriterion path problems. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making*: *Theory and Applications*, pages 109–127. Lecture Notes in Economics and Mathematical Systems, 177. Springer-Verlag, New York, 1980.

[8] R. Hassin and A. Levin. An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM Journal on Computing*, 33(2):261–268, 2004.

[9] S. P. Hong, S. J. Chung, and B. H. Park. A fully-polynomial bicriteria approximation scheme for the constrained minimum spanning tree problem. *Operations Research Letters*, 32(3):233–239, 2004.

[10] E. Horowitz and S. Sahni. Exact and approximate algorithms for scheduling non-identical processors. *Journal of the ACM*, 23:317–327, 1976.

[11] K. Jansen and L. Porkolab. Improved approximation schemes for scheduling unrelated parallel machines. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 408–417, 1999.

[12] D. R. Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.

[13] D. R. Karger. Minimum cuts in near-linear time. *Journal of the ACM*, 47(1):46–76, 2000.

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

[15] J. K. Lenstra, D. B. Shmoys, and E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46:259–271, 1990.

[16] H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Mathematics*, 5(1):54–66, 1992.

[17] H. Nagamochi, K. Nishimura, and T. Ibaraki. Computing all small cuts in an undirected network. *SIAM Journal on Discrete Mathematics*, 10(3):469–481, 1997.

[18] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 86–92, 2000.

[19] K. Schloegel, G. Karypis, and V. Kumar. A new algorithm for multi-objective graph partitioning. In *Proceedings of the European Conference on Parallel Processing*, pages 322–331, 1999.

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