Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Lehrstuhl für Angewandte Mathematik
und Numerische Mathematik

Lehrstuhl für Optimierung und Approximation

Preprint BUW-AMNA-OPAP 10/20

Jochen Gorski, Kathrin Klamroth and Stefan Ruzika

Generalized Multiple Objective Bottleneck Problems

November 2010

http://www2.math.uni-wuppertal.de

Generalized Multiple Objective Bottleneck Problems

Jochen Gorski[a], Kathrin Klamroth[a,*], Stefan Ruzika[b]

[a] *University of Wuppertal, Faculty of Mathematics and Natural Sciences, Department of Mathematics and Informatics, Gaußstr. 20, 41097 Wuppertal, Germany.*
[b] *University of Kaiserslautern, Department of Mathematics, P.O.Box 3049, Paul-Ehrlich-Str. 14, 67653 Kaiserslautern, Germany.*

Abstract

We consider multiple objective combinatorial optimization problems in which the first objective is of arbitrary type and the remaining objectives are either bottleneck or k-max objective functions. While the objective value of a bottleneck objective is determined by the largest cost value of any element in a feasible solution, the k^{th}-largest element defines the objective value of the k-max objective. An efficient solution approach for the generation of the complete nondominated set is developed which is independent of the specific combinatorial problem at hand. This implies a polynomial time algorithm for several important problem classes like shortest paths, spanning tree, and assignment problems with bottleneck objectives which are known to be $\mathcal{NP}$-hard in the general multiple objective case.

Keywords: combinatorial optimization, multiple objective, bottleneck, k-max

1. Introduction

Combinatorial bottleneck problems have applications, for example, in project management, engineering design, traffic management, and software engineering. One or several *bottleneck processes* or *bottleneck components* with restricted capacity or throughput limit the performance of the overall system, and thus solutions are sought which minimize this bottleneck (*min-max problems*). In this article we consider combinatorial optimization problems with multiple objectives, one or several of which are *bottleneck objectives* or *k-max objectives* that evolve as a natural generalization of bottleneck objectives.

More precisely, let $\mathcal{E} = \{e_1, \ldots, e_n\}$ be a finite set of $n \in \mathbb{N}$ different elements, and let $\mathcal{X} \subseteq \mathcal{P}(\mathcal{E})$, where $\mathcal{P}(\mathcal{E})$ denotes the power set of $\mathcal{E}$. Furthermore, let $c\colon \mathcal{E} \to \mathbb{Z}$ denote a scalar cost function on the elements of $\mathcal{E}$. Then,

$$b(S) = \max_{e \in S}\{c(e)\}$$

defines a *bottleneck objective function* yielding the maximum cost coefficient of a feasible solution $S \in \mathcal{X}$. Minimizing a single bottleneck objective over the complete feasible set $\mathcal{X}$ is called a *combinatorial bottleneck problem* (CBP). This problem is also frequently called *min-max problem* in the literature.

Now, let $|S| \geq m$ for some $m \in \{1, \ldots, n = |\mathcal{E}|\}$ and all $S \in \mathcal{X}$ and let $k \in \{1, \ldots, m\}$ be arbitrary but fixed. Then, a *k-max objective function* is defined by

$$g(S) = k\text{-}\max_{e \in S}\{c(e)\},$$

returning the k^{th} largest cost coefficient among the elements of S (cf. also [1]). The corresponding minimization problem over $\mathcal{X}$ is referred to as the *k-max optimization problem* (kMAX). For the case $k = 1$,

*Corresponding author.
Email address: klamroth@math.uni-wuppertal.de (Kathrin Klamroth)

the bottleneck and k-max objective coincide. Hence, Problem (kMAX) can be seen as a generalized version of Problem (CBP). To clarify notation, we introduce the following convention concerning the notion of a k-max objective: If we refer to an optimization problem involving at least one k-max objective, we speak of a k-max problem and of a k-max objective. If a special instance of the objective is considered for a fixed $k \in \{1, \ldots, m\}$, we write $k\text{-max}_{e \in S}\{c(e)\}$ or equivalently $k\text{-max}(S)$.

In the following, let $c^1, \ldots, c^p \colon \mathcal{E} \to \mathbb{Z}$ denote p different cost functions on the set $\mathcal{E}$ and let $f \colon \mathcal{X} \to \mathbb{Z}$ denote an additional objective function of arbitrary type. Furthermore, let the functions $f_i \colon \mathcal{X} \to \mathbb{Z}$, $i \in I_p$, correspond either to p bottleneck or p k-max objectives, where $I_p = \{1, \ldots, p\}$. Then, a *multiple objective combinatorial optimization problem* (MCOP) is given by

$$\begin{array}{ll} \min & F(S) = (f(S), f_1(S), \ldots, f_p(S))^\top \\ \text{s.t.} & S \in \mathcal{X}, \end{array} \tag{MCOP}$$

where $F \colon \mathcal{X} \to \mathbb{Z}^{p+1}$ consists of $(p+1)$ integer valued objective functions. In this context, the set $\mathcal{E}$ is called *ground set*, the set $\mathcal{X}$ *feasible set* in the *decision space* and $\mathcal{Y} = F(\mathcal{X})$ denotes the *set of attainable outcomes* in the *outcome space*. Any $S \in \mathcal{X}$ is called a *feasible solution*. An instance of (MCOP) is denoted by $(\mathcal{E}, \mathcal{X}, F)$. Neglecting the objectives $f_1, \ldots, f_p$ defines a single objective instance $(\mathcal{E}, \mathcal{X}, f)$. If the involved objective functions f_i correspond to p bottleneck objectives, we refer to (MCOP) as a *multiple objective combinatorial bottleneck problem* (MCBP). For the case of p k-max objectives, (MCOP) is called *multiple objective k-max optimization problem* (MkMAX). As in the single objective case, (MCBP) is a special case of (MkMAX), where (MCBP) can be obtained by choosing $k_1 = \ldots = k_p = 1$.

Since there does not exist a canonical ordering on the Euclidean vector space $\mathbb{R}^{p+1}$ if $p \geq 1$, we use the following concepts of *componentwise orderings*:

$$\begin{array}{lcl} y^1 \leqq y^2 & \iff & y^1_i \leq y^2_i,\ i = 1, \ldots, p+1, \\ y^1 \leq y^2 & \iff & y^1_i \leq y^2_i,\ i = 1, \ldots, p+1 \text{ and } y^1 \neq y^2, \\ y^1 < y^2 & \iff & y^1_i < y^2_i,\ i = 1, \ldots, p+1. \end{array}$$

Following the Pareto concept of optimality, we say that a feasible point $S^1 \in \mathcal{X}$ *dominates* a point $S^2 \in \mathcal{X}$ if $F(S^1) \leq F(S^2)$. If strict inequality holds for all p components, i.e., $F(S^1) < F(S^2)$, then S^1 *strongly dominates* S^2. If there does not exist any feasible point which dominates $S \in \mathcal{X}$, we say that S is an *efficient solution* of (MCOP). For the case that there exists no feasible point that strongly dominates $S \in \mathcal{X}$, we say that S is *weakly efficient* for (MCOP). If there is no $S^2 \in \mathcal{X}$, $S^2 \neq S^1$, such that $F(S^2) \leqq F(S^1)$, S^1 is called *strictly efficient*. Note that strict efficiency is the multiple objective analogon to unique optimal solutions for single objective problems. According to these definitions, the *efficient set* $\mathcal{X}_E$ is defined by

$$\mathcal{X}_E = \{S \in \mathcal{X} : \text{there exists no } \bar{S} \in \mathcal{X} \text{ with } F(\bar{S}) \leq F(S)\}.$$

Using the vector-valued mapping F, the image of $\mathcal{X}_E$ is called the *non-dominated set* $\mathcal{Y}_N$. In this context, a point $y^2 \in \mathbb{R}^{p+1}$ is called *dominated* by $y^1 \in \mathbb{R}^{p+1}$ if $y^1 \leq y^2$. Since the cardinality of the efficient set may be of exponential size even though the size of the non-dominated set may be known to be bounded by a polynomial, one is often interested in determining the complete set of non-dominated solutions $\mathcal{Y}_N$ and a corresponding efficient representative from $\mathcal{X}_E$, rather than the complete set $\mathcal{X}_E$ itself. In this context, any subset $\mathcal{X}' \subseteq \mathcal{X}$ is called a *complete set* of efficient solutions if $f(\mathcal{X}') = \mathcal{Y}_N$ holds. Two efficient solutions $S^1, S^2 \in \mathcal{X}_E$ are called *equivalent* if $F(S^1) = F(S^2)$. In addition, an efficient solution is called *supported* if it can be obtained by optimizing a linear combination of the objectives involved with non-negative weights over the feasible set $\mathcal{X}$.

As shown in [2], given an appropriately chosen right-hand side vector $\varepsilon \in \mathbb{R}^p$, solving the *$\varepsilon$-constraint problem*

$$\begin{array}{rll} \min & f(S) & \\ \text{s.t.} & f_i(S) \leq \varepsilon_i, & i = 1, \ldots, p, \\ & S \in \mathcal{X} & \end{array} \tag{1}$$

yields a solution which is weakly efficient for (MCOP). Moreover, a complete set of efficient solutions of (MCOP) and the corresponding non-dominated set can be generated by solving a sequence of appropriate ε-constraint problems. However, the number of problems which have to be solved to compute a complete set of efficient solutions may grow exponentially in the size of the input data.

The remainder of this article is organized as follows. In Section 2 we present a detailed literature review for solving single objective bottleneck as well as k-max optimization problems. In addition, we summarize the results for the multiple objective bottleneck problem obtained for special classes of combinatorial optimization problems. In Section 3 we present an algorithm that can be applied to solve both (MCBP) and (MkMAX) based on sequential ε-constraint scalarizations. For (MCBP) we discuss an approach that solves Problem (1) independently from any special class of combinatorial problem or the number of involved bottleneck objectives. Based on the ideas developed in [1] for the single objective case, we re-model Problem (1) for (MkMAX). This new model allows to solve Problem (1) more efficiently. We finally conclude the article in Section 4 and present some more new research ideas related to the topic.

2. Literature Review

In the following, we briefly review the existing literature for solving (special classes of) combinatorial bottleneck problems as well as single objective k-max optimization problems.

Frequently, combinatorial bottleneck problems with a single objective are solved by using the so-called *threshold algorithm* that further builds the basis for a solution approach derived for the multiple objective case in the next section. The idea of using this approach was already introduced by Edmonds and Fulkerson [3] in the context of bottleneck extrema and was adopted by many other authors. Given a feasible solution $S \in \mathcal{X}$, all elements $e \in \mathcal{E}$ satisfying $c(e) \geq b(S)$ are deleted from the ground set $\mathcal{E}$, and a new feasible solution is determined. If such a solution $S' \in \mathcal{X}$ exists, $b(S')$ can be seen as a threshold value for the optimal objective value. All elements $e \in \mathcal{E}$ satisfying $c(e) \geq b(S')$ can be removed from the ground set and the feasibility problem is resolved. This procedure is iteratively repeated until no further feasible solution can be found. Then, the last solution that has been feasible for the modified problem is also optimal for (CBP) and its cost is the optimal objective value. Applying this procedure the threshold approach solves Problem (CBP) in $\mathcal{O}(T \log(|\mathcal{E}|))$ where T denotes the time to solve the corresponding feasibility problem. However, there exist classes of combinatorial problems in which the bottleneck problem is proven to be $\mathcal{NP}$-complete in general (see, e.g., [4] for the traveling salesman problem).

While for many combinatorial problems on graphs and networks, like the shortest path, spanning tree or the traveling salesman problem, it is quite easy to update the feasible set automatically by deleting those nodes or edges whose costs exceed the current threshold value, there also exist combinatorial problems in which the feasible set $\mathcal{X}$ cannot be updated in a straightforward and efficient way. Hence, discarding infeasible solutions from $\mathcal{X}$ may not be an easy task. In this case, the threshold approach combined with a cost modification scheme can be used to solve (CBP). Instead of deleting those elements from the ground set that exceed the current threshold value, the corresponding cost coefficients are set to infinity (cf. also [5]) and thus, solutions containing such elements are penalized.

Note that in the latter article, an alternative non-parametric transformation from (CBP) to a combinatorial problem with sum objective was suggested that only depends on the number of (different) cost coefficients of c. In the early 1980s it was shown in [6] that an optimal solution of (CBP) can be computed by solving an associated combinatorial problem with sum objective if an appropriate transformation of the given cost function c is applied. Based on certain properties that have to be satisfied by such a transformation, a whole class of suitable transformations is derived in this article.

The multiple objective bottleneck problem was studied by several authors for special classes of combinatorial problems involving at most three different bottleneck objectives. In the sequence of articles [7–10] the subset of supported non-dominated solutions among the set of non-dominated solutions was numerically analyzed for the linear assignment, the minimum spanning tree, the asymmetric traveling salesman and the knapsack problem with either two or three bottleneck objectives or one sum and one or two bottleneck objectives.

Besides these articles there exist some publications that deal with the multiple objective shortest path problem. Polynomial algorithms considering a sum and a bottleneck objective are presented in [11–14]. More recently, shortest path problems involving two bottleneck and a sum objective were analyzed in [15] and [16]. The authors solved the triobjective problem by a similar approach as the one that we will present for an arbitrary number of bottleneck objectives in Section 3.1. Furthermore, an efficient handling of the bounds on the sequence of ε-constraint problems that has to be solved to generate a complete set of efficient solutions is discussed in [16]. An improved and generalized version of this approach can be found in Section 3.

Algebraic sum problems involving a sum and a bottleneck objective are discussed in [17] and [18]. The authors propose algorithms based on a biobjective approach that can be seen as special versions of the general algorithm presented in Section 3.1. Finally, the notion of a k-max objective was already used in [19–21] in the context of lexicographic bottleneck problems, and in [22] where lexicographic balanced optimization problems are discussed. However, k-max optimization problems were discussed in [1] for the first time explicitly. The authors present a bisection algorithm that is based on iteratively solving an associated sum objective problem with binary cost coefficients, applicable to general combinatorial optimization problems as defined in Section 1. A time bound of $\mathcal{O}(T \log(|\mathcal{E}|))$ is stated for this approach, where T denotes the time to solve the sum problem with binary costs. We further utilize the ideas presented in [1] to derive an ε-constraint approach for Problem (kMAX) in Section 3.2.

3. The Algorithms

In this section we present algorithms that generate a complete set of efficient solutions for the multiple objective bottleneck as well as k-max problems. As (MCBP) and (MkMAX) share many properties, we discuss the proposed solution approach in a generalized framework at the beginning of this section, while we go into futher details with respect to the involved ε-constraint subproblems in Sections 3.1 and 3.2 for the bottleneck and k-max case, respectively.

Let $f\colon \mathcal{X} \to \mathbb{Z}$ denote an objective of arbitrary type, while the functions $f_i\colon \mathcal{X} \to \mathbb{Z}$, $i \in I_p$, correspond either to p bottleneck or p k-max objectives as defined in Section 1. We consider an approach to solve (MCOP) that is induced by solving a sequence of ε-constraint problems. Note that for fixed $\varepsilon \in \mathbb{R}^p$ the non-dominated solution that is obtained by solving Problem (1) crucially depends on the chosen right hand side components $\varepsilon_i \in \mathbb{R}$, $i \in I_p$. In the worst case, an exponential number of ε-constraint problems has to be solved to determine a complete set of efficient solutions.

More information is available if problems of the form (MCBP) and (MkMAX) are considered. Since for each $i \in \{1, \dots, p\}$ and $S \in \mathcal{X}$, $f_i(S)$ takes at most n distinct values contained in the set $c^i(\mathcal{E}) := \{c^i(e) : e \in \mathcal{E}\}$, all right hand side values are known in advance. Furthermore, their number is linearly bounded by n. Since for each combination of right hand side components ε_i, at most one non-dominated solution exists, this implies the following result:

Lemma 3.1 *The cardinality of the non-dominated set of both (MCBP) and (MkMAX) is bounded by $\mathcal{O}(n^p)$.*

Based on this observation, we derive a solution approach which determines a complete set of efficient solutions for both problems (MCBP) and (MkMAX): we iteratively solve Problem (1) for all combinations of right hand side values contained in the sets $c^i(\mathcal{E})$, $i \in I_p$, followed by a filter step to exclude weakly efficient solutions that are not contained in the efficient set of the given problem.

To minimize the number of ε-constraint problems to be solved, we apply an efficient scheme to handle the bounds that have to be set on the constraint functions $f_1, \dots, f_p$. Let $c^i : \mathcal{E} \to \mathbb{Z}$ denote the cost function associated to the objective f_i, $i = 1 \dots, p$. Furthermore, let $m_i \in \{1, \dots, n\}$ correspond to the number of different values contained in $c^i(\mathcal{E})$ and suppose that these are arranged in decreasing order, i.e. $c^i_1 > c^i_2 > \dots > c^i_{m_i}$ for $i \in I_p$. We set $\mathcal{A} = \{\nu \in \mathbb{R}^p : \nu_i \in \{1, \dots, m_i\},\ i \in I_p\}$ and consider the ε-constraint problem

$$\begin{aligned} \min\ & f(S) \\ \text{s.t. } & f_i(S) \leq c^i_{\nu_i},\ i = 1, \dots, p, \\ & S \in \mathcal{X}, \end{aligned} \tag{A_ν}$$

Algorithm 1 Threshold Algorithm for (MCOP)

Input: An instance $(\mathcal{E}, \mathcal{X}, (f, f_1, \ldots, f_p))$ of (MCOP)
Output: A complete subset $\mathcal{X}^\star \subseteq \mathcal{X}_E$.

Set $\mathcal{X}^\star = \emptyset$ and $A = \mathbf{1} \in \mathbb{R}^{m_1 \times \ldots \times m_p}$ with index set $\mathcal{A}$.
while $A \neq \mathbf{0}$ **do**
  Determine $\bar{\nu} = \text{lexmin}\{\nu \in \mathcal{A} : a_\nu \neq 0\}$.
  Call Algorithm 2 or Algorithm 3 to solve Problem $(A_{\bar{\nu}})$, respectively.
  if Problem $(A_{\bar{\nu}})$ is feasible with optimal solution $S^\star$ **then**
    Determine $\nu^\star \in \mathcal{A}$, s.t. $f_i(S^\star) = c^i_{\nu^\star_i}$, $i \in I_p$.
    Set $a_{\tilde{\nu}} = 0$ for all $\tilde{\nu} \in \{\nu \in \mathcal{A} : \bar{\nu} \leqq \nu \leqq \nu^\star\}$.
    Update $\mathcal{X}^\star = \mathcal{X}^\star \cup \{S^\star\}$.
  else
    Set $a_{\tilde{\nu}} = 0$ for all $\tilde{\nu} \in \{\nu \in \mathcal{A} : \bar{\nu} \leqq \nu\}$.
  end if
end while
Filter $\mathcal{X}^\star$ for equivalent and dominated solutions.
return $\mathcal{X}^\star$.

where $\nu \in \mathcal{A}$. We state:

Lemma 3.2 *Let $\bar{\nu} \in \mathcal{A}$. Let $S^\star$ be an optimal solution of Problem $(A_{\bar{\nu}})$ and let $\nu^\star \in \mathcal{A}$ such that $f_i(S^\star) = c^i_{\nu^\star_i}$ for $i \in I_p$. Then $S^\star$ is also optimal for Problem (A_ν) for all $\nu \in \mathcal{A}$ satisfying $\bar{\nu} \leqq \nu \leqq \nu^\star$.*

Proof. If $\bar{\nu} = \nu^\star$, there is nothing to show. Otherwise, assume there exists $\nu \in \mathcal{A}$ such that $\bar{\nu} \leq \nu \leqq \nu^\star$ and $S^\star$ is not optimal for Problem (A_ν), i.e. there exists $S \in \mathcal{X}$ satisfying $f(S) < f(S^\star)$. However, since S is feasible for Problem (A_ν), this implies that $f_i(S) \leq c^i_{\nu_i} < c^i_{\bar{\nu}_i}$ for all $i \in I_p$ as the c^i's are sorted in decreasing order. Hence, S is also feasible for Problem $(A_{\bar{\nu}})$. This contradicts the optimality of $S^\star$ for Problem $(A_{\bar{\nu}})$. □

Lemma 3.2 implies that $S^\star$, as an optimal solution of Problem $(A_{\bar{\nu}})$, yields a lower bound on the optimal function value of Problem (A_ν) for all $\nu \in \mathcal{A}$ satisfying $\bar{\nu} \leq \nu$. This fact can be of interest whenever Problem (A_ν) is solved by applying an algorithm that is based on a branch and bound approach. For the case that Problem $(A_{\bar{\nu}})$ is infeasible, we state:

Lemma 3.3 *Let $\bar{\nu} \in \mathcal{A}$. If Problem $(A_{\bar{\nu}})$ is infeasible, then Problem (A_ν) is also infeasible for all $\bar{\nu} \in \mathcal{A}$ satisfying $\bar{\nu} \leqq \nu$.*

Proof. If $\bar{\nu} = \nu$, there is nothing to show. Otherwise, assume there exists $\nu \in \mathcal{A}$ such that $\bar{\nu} \leq \nu$ and Problem (A_ν) is feasible, i.e. there exists $S \in \mathcal{X}$ satisfying $f_i(S) \leq c^i_{\nu_i}$ for all $i \in I_p$. But since $\bar{\nu} \leq \nu$ and the c^i's are sorted in decreasing order, this further implies that $f_i(S) \leq c^i_{\bar{\nu}_i}$ for all $i \in I_p$, i.e. S is also feasible for Problem $(A_{\bar{\nu}})$ which yields a contradiction. □

Applying the results of Lemma 3.2 and Lemma 3.3 we can derive an efficient scheme that efficiently reduces the number of subproblems that have to be solved to calculate a complete set of efficient solutions for (MCOP). For multidimensional matrices $A, Z \in \mathbb{R}^{m_1 \times \ldots \times m_p}$, let $A = \mathbf{1}$ and $Z = \mathbf{0}$ denote the matrices having coefficients of ones and zeros only, i.e. $a_\nu = 1$ and $z_\nu = 0$ for all $\nu \in \mathcal{A}$. We use the coefficients of the matrix A as an indicator whether the corresponding ε-constraint problem that is uniquely defined by the position of a coefficient in A has to be solved or not. For $\nu \in \mathcal{A}$, $a_\nu = 1$ means that Problem (A_ν) has still to be solved during the course of the algorithm, while a_ν replaced by the corresponding entry z_ν implies that Problem (A_ν) does not have to be solved due to Lemmas 3.2 and 3.3.

An algorithmic description of our approach can be found in Algorithm 1. Depending on the given optimization problem (MCBP) or (MkMAX), either Algorithm 2 or Algorithm 3, described in Subsection 3.1 and Subsection 3.2, is called to solve Problem (A_ν). As an ε-constraint approach is used to solve the given

optimization problem, a filtering for equivalent and dominated solutions has to be performed at the end of the algorithm. This can be done, e.g., as described in [16] for the triobjective shortest path problem with two bottleneck objectives.

Theorem 3.4 *Algorithm 1 is correct and solves (MCOP) in $\mathcal{O}(n^p \cdot T)$, where n is the number of elements contained in the ground set, p is the number of involved bottleneck or k-max objectives and T denotes the time to solve the associated subproblems applying Algorithm 2 and Algorithm 3, respectively.*

This result shows that the class of multiple objective combinatorial optimization problems with one general and several bottleneck and k-max objectives can be solved in polynomial time (if T is polynomial) even though this is in general not possible for the corresponding multiple objective combinatorial optimization problems with more than one general objective function. This property of (MCBP) and (MkMAX) will be further analyzed in the subsequent subsections.

3.1. Multiple Objective Bottleneck Problems

We now focus on the efficient solution of the ε-constraint problem (A_ν) in the case of bottleneck objectives. The method suggested generalizes the solution approaches for some combinatorial problems involving at most two bottleneck objectives reviewed in Section 2. In contrast to these articles, we present an algorithm that can be applied to any combinatorial optimization problem with an arbitrary number of bottleneck objectives and an additional objective function f of arbitrary type. In the following we discuss the special properties of Problem (A_ν) for the case that the objectives $f_1, \ldots, f_p$ correspond to p bottleneck objectives $b_1, \ldots, b_p$.

Let the right hand side of Problem (A_ν) be given by arbitrary but fixed values $c^i_{\nu_i}$ for $\nu \in \mathcal{A}$ and $i \in I_p$. We consider the i^{th} constraint which is formally given by $b_i(S) \leq c^i_{\nu_i}$. Obviously, this constraint implies that a solution $S \in \mathcal{X}$ is feasible if and only if its maximum cost coefficient with respect to the cost function c^i is at most $c^i_{\nu_i}$. Hence, as for the single objective case, $c^i_{\nu_i}$ can be treated as a threshold value for the cost coefficients contained in the set $c^i(\mathcal{E})$.

Since for many combinatorial problems the feasible set $\mathcal{X}$ is not given explicitly, the elimination of all elements from the ground set satisfying $c^i(e) > c^i_{\nu_i}$ (to enforce the constraint $b_i(S) \leq c^i_{\nu_i}$) may not be conducive. As in the single-objective case, we therefore suggest a slightly different approach for the solution of Problem (A_ν). We assume in the following that the objective function f depends on a given cost function w on the elements of the ground set $\mathcal{E}$. To indicate this dependency, we replace f by f_w.

Let $\nu \in \mathcal{A}$ be arbitrary but fixed. Instead of removing the elements from $\mathcal{E}$ that do not satisfy the constraint $b_i(S) \leq c^i_{\nu_i}$, we define a modified cost function $\tilde{w} : \mathcal{E} \to w(\mathcal{E}) \cup \{+\infty\}$, where

$$\tilde{w}(e) = \begin{cases} w(e), & \text{if } c^i(e) \leq c^i_{\nu_i} \quad \forall i \in \{1, \ldots, p\}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{2}$$

Based on this modified cost function $\tilde{w}$, we solve the unconstrained single objective combinatorial optimization problem which is given by $(\mathcal{E}, \mathcal{X}, f_{\tilde{w}})$.

Lemma 3.5 *If the optimal objective value of the instance $(\mathcal{E}, \mathcal{X}, f_{\tilde{w}})$ is finite, the solution obtained is both feasible as well as optimal for Problem (A_ν) with respect to the considered right hand side values $c^i_{\nu_i}$, where $i \in I_p$. Otherwise, Problem (A_ν) is infeasible.*

Proof. Let S denote the optimal solution of the instance $(\mathcal{E}, \mathcal{X}, f_{\tilde{w}})$, and let $f_{\tilde{w}}(S) < \infty$. By construction of $\tilde{w}$ we have that $c^i(e) \leq c^i_{\nu_i}$ for all $e \in S$ and $i \in I_p$. Hence, S is feasible for Problem (A_ν) and $f_{\tilde{w}}(S) = f_w(S)$. Suppose that S is not optimal for Problem (A_ν). Then there exists $S^\star \in X$ with $f_w(S^\star) < f_w(S)$ satisfying $b_i(S^\star) \leq c^i_{\nu_i}$ for all $i \in I_p$ which means that $c^i(e) \leq c^i_{\nu_i}$ for all $e \in S^\star$ and $i \in I_p$. Hence, $f_{\tilde{w}}(S^\star) = f_w(S^\star) < f_w(S) = f_{\tilde{w}}(S)$ which contradicts the optimality of S for the instance $(\mathcal{E}, \mathcal{X}, f_{\tilde{w}})$.

On the other hand, if the optimal objective value for the instance $(\mathcal{E}, \mathcal{X}, f_{\tilde{w}})$ is not finite, there does not exist any feasible solution satisfying $c^i(e) \leq c^i_{\nu_i}$ for all $i \in I_p$ simultaneously. Hence, Problem (A_ν) is infeasible. □

Algorithm 2 Algorithm for solving Problem (A_ν) for (MCBP)

Input: An instance $(\mathcal{E}, \mathcal{X}, (f_w, b_1, \ldots, b_p))$ of (MCBP) and an index vector $\nu \in \mathcal{A}$.
Output: An optimal solution $S^\star$ of Problem (A_ν).

```
for all e ∈ E do
    if c^i(e) ≤ c^i_{ν_i} for all i ∈ {1, ..., p} then
        Set w̃(e) = w(e).
    else
        Set w̃(e) = +∞.
    end if
end for
Solve the unconstrained problem (E, X, f_w̃) → S*.
return S*.
```

As a result of Lemma 3.5, Problem (A_ν) can be solved without explicitly deleting elements from the ground set $\mathcal{E}$. After a simple modification of the costs of the given cost function (2), an unconstrained problem with objective $f_{\tilde{w}}$ has to be solved. This approach is summarized in Algorithm 2.

Corollary 3.6 *Algorithm 1 in combination with Algorithm 2 is correct and solves (MCBP) in $O(n^p \cdot T)$, where n is the number of elements contained in the ground set, p is the number of bottleneck objectives and T denotes the time to solve the unconstrained combinatorial problem with objective function $f_{\tilde{w}}$.*

Proof. The result follows immediately from Theorem 3.4 and Lemma 3.5. □

Theorem 3.4 implies that (MCBP) is solvable in polynomial time if the single objective problem $(\mathcal{E}, \mathcal{X}, f_w)$ is. For the case that the objective f_w corresponds to a sum objective, this property is satisfied for many classes of combinatorial problems like shortest path, spanning tree and assignment problems, to mention only some examples.

3.2. Multiple Objective k-max Problems

Analogous to Subsection 3.1 we are interested in solving Problem (A_ν) where $\nu \in \mathcal{A}$ for the case that the p given objective functions correspond to the k-max objectives $g_1, \ldots, g_p$. In contrast to the bottleneck case there is no obvious way to handle the side constraints $g_i(S) = k_i\text{-max}(S) \leq c^i_{\nu_i}$ for the case that $k_i > 2$ efficiently, since elements $e \in \mathcal{E}$ with $c(e) > c^i_{\nu_i}$ may be contained in a feasible solution of Problem (A_ν). To overcome this difficulty, we apply a similar auxiliary construction as it is used to solve the single objective k-max problem presented in [1]. In more detail, we relate each k-max objective to a binary sum objective.

Let $\nu \in \mathcal{A}$ and $i \in I_p$. We use $c^i_{\nu_i} \in c^i(\mathcal{E})$ as a threshold value and assign binary weights to each element $e \in \mathcal{E}$ by setting

$$w_{\nu_i}(e) = \begin{cases} 0, & \text{if} \quad c^i(e) \leq c^i_{\nu_i}, \\ 1, & \text{if} \quad c^i(e) > c^i_{\nu_i}. \end{cases}$$

The new i^{th} auxiliary function $f_{\nu_i} : \mathcal{X} \to \mathbb{N}$ is then given by

$$f_{\nu_i}(S) = \sum_{e \in S} w_{\nu_i}(e).$$

Instead of solving Problem (A_ν) directly, we consider the auxiliary problem

$$\begin{aligned} \min \ & f(S) \\ \text{s.t.} \ & f_{\nu_i}(S) \leq k_i - 1, \quad i = 1, \ldots, p, \\ & S \in \mathcal{X}, \end{aligned} \tag{B_ν}$$

where $\nu_i \in \{1, \ldots, m_i\}$ is uniquely determined by the right hand side value $c^i_{\nu_i}$ of Problem (A_ν) for all $i \in I_p$.

Algorithm 3 Algorithm for solving Problem (A_ν) for (MkMAX)

Input: An instance $(\mathcal{E}, \mathcal{X}, (f, g_1, \ldots, g_p))$ of (MkMAX) and an index vector $\nu \in \mathcal{A}$.
Output: An optimal solution $S^\star$ of Problem (A_ν).

```
for All e ∈ E do
    for All i = 1 to p do
        if c^i(e) ≤ c^i_{ν_i} then
            Set w_{ν_i}(e) = 0.
        else
            Set w_{ν_i}(e) = 1.
        end if
    end for
end for
Solve the auxiliary problem given by Problem (B_ν) → S*.
return S*.
```

Problem (B_ν) can be interpreted as follows: For each $i \in I_p$, the side constraint $f_{\nu_i}(S) \leq k_i - 1$ introduces an additional knapsack constraint to the single objective combinatorial optimization problem with objective function f. These side constraints can be seen as counters on the cardinality of the most expensive elements from $\mathcal{E}$ with respect to the given cost functions $c_1, \ldots, c_p$ which are contained in a feasible solution. For fixed ν_i only up to $k_i - 1$ of these most expensive elements with respect to the cost function c^i are allowed to be included in a feasible solution, where threshold value $c^i_{\nu_i}$ indicates which elements have to be considered as "expensive" and which not. Hence, Problem (B_ν) can further be seen as a multiple choice version of the considered combinatorial optimization problem. We state:

Theorem 3.7 *Problem (A_ν) is infeasible if and only if Problem (B_ν) is infeasible. $S^\star$ is optimal for Problem (A_ν) if and only if it is optimal for Problem (B_ν).*

Proof. Since the objectives of Problem (A_ν) and Problem (B_ν) coincide, it suffices to show that $S \in \mathcal{X}$ is feasible for Problem (A_ν) if and only if S is feasible for Problem (B_ν). So, let S be feasible for Problem (A_ν), i.e.

$$g_i(S) \;=\; k_i\text{-}\max_{e \in S}\{c^i(e)\} \;\leq\; c^i_{\nu_i} \text{ for all } i \in I_p.$$

This implies that at maximum $k_i - 1$ elements $e \in S$ satisfy $c^i(e) > c^i_{\nu_i}$, i.e. $f_{\nu_i}(S) = \sum_{e \in S} w_{\nu_i}(e) \leq k_i - 1$ for all $i \in I_p$. Hence, S is also feasible for Problem (B_ν).

Conversely, let S be feasible for Problem (B_ν). For fixed $i \in I_p$ this implies that $f_{\nu_i}(S) = \sum_{e \in S} w_{\nu_i}(e) \leq k_i - 1$, and S contains at maximum $k_i - 1$ elements $e \in \mathcal{E}$ satisfying $c^i(e) > c^i_{\nu_i}$ for all $i \in I_p$. Hence, $g_i(S) = k_i\text{-}\max(S) \leq c^i_{\nu_i}$, and S is also feasible for Problem (A_ν). □

Theorem 3.7 implies that we can find an optimal solution for Problem (A_ν) by solving Problem (B_ν). Furthermore, Theorem 3.7 can be used to directly relate optimal solutions of Problem (B_ν) to efficient solutions of (MkMAX) and vice versa. In order to prove efficiency for optimal solutions of Problem (B_ν), uniqueness of the optimal solution for this problem yields a sufficient condition (cf. [2]). From the results in [2] for the general ε-constraint problem it follows:

Corollary 3.8 *It holds:*

1. *Let S_ν be an optimal solution of Problem (B_ν) for some $\nu \in \mathcal{A}$. Then S_ν is a weakly efficient solution of (MkMAX).*
2. *Let S_ν be a unique optimal solution of Problem (B_ν) for some $\nu \in \mathcal{A}$. Then S_ν is a strictly efficient solution of (MkMAX).*
3. *If $S^\star$ is an efficient solution of (MkMAX), then there exists $\nu \in \mathcal{A}$ such that $S^\star$ is optimal for Problem (B_ν).*

The algorithmic consequence of Theorem 3.7 for $(MkMAX)$ is summarized in Algorithm 3. The algorithm computes an optimal solution for Problem (A_ν) by means of Problem (B_ν). To shorten the algorithmic presentation, the case that Problem (B_ν) is infeasible is not treated separately in Algorithm 3. It is assumed that Algorithm 3 returns a special solution $S^\star$ indicating that Problem (B_ν) and hence Problem (A_ν) is infeasible. Similar to Corollary 3.6 we state:

Theorem 3.9 *Algorithm 1 in combination with Algorithm 3 correctly determines the set of non-dominated solutions $\mathcal{Y}_N$ of (MkMAX) in $\mathcal{O}(n^p \cdot T)$, where T denotes the maximum time for solving the Problem (B_ν) in each iteration of the algorithm.*

Proof. The result follows immediately from Theorem 3.4 in combination with Theorem 3.7. □

Theorem 3.9 states a theoretical time bound that is independent from any special class of combinatorial problems. Since constrained versions of polynomially solvable single objective combinatorial optimization problems are often $\mathcal{NP}$-hard problems (even if the minimum spanning tree (cf. [23]) or the shortest-path problem (cf. [4]) is considered), Problem (B_ν) is $\mathcal{NP}$-hard to solve in general. However, due to the simple binary structure of the auxiliary sum objectives f_{ν_i} $(i \in I_p)$ it may be possible to derive algorithms that can solve this special type of problem in polynomial time depending on the considered class of combinatorial problems. In this case, also (MkMAX) is solvable in polynomial time due to Theorem 3.9.

To give an example, we consider the biobjective minimum spanning tree problem, where one of the two objective functions is given as a k-max objective. Let a connected graph $G = (V, A)$ be given, where V and A denote the set of nodes and the set of edges of G, respectively. Furthermore, let f and $g_1 = k_1\text{-max}(S)$ denote an arbitrary sum objective and a k-max objective, respectively, where $k_1 \in \{2, \ldots, |V| - 1\}$. To solve Problem (B_ν) efficiently, we use the algorithm proposed in [24] to solve matroid intersection problems. The authors stated a time bound of $\mathcal{O}(m + n\log(n))$ for their approach, where $|V| = n$ and $|A| = m$. Applying this result to Theorem 3.9 we conclude:

Corollary 3.10 *The biobjective minimum spanning tree problem on a connected graph $G = (V, A)$ with a sum objective and a k-max objective can be solved in $\mathcal{O}(m^2 + mn\log(n))$, where $|V| = n$ and $|A| = m$.*

4. Conclusions and Future Research

In this article we considered general multiple objective combinatorial optimization problems with one arbitrary and one or several additional bottleneck or k-max objective functions. An efficient solution approach based on the iterative solution of ε-constraint scalarizations was presented which is independent of the specific combinatorial problem considered. The solution of the resulting subproblems is discussed both for the bottleneck and the k-max case, leading to a polynomial time algorithm for several important cases like shortest paths, spanning tree and assignment problems in the bottleneck case, and spanning tree problems in the k-max case.

The k-max case is particularly interesting since the involved ε-constraint problems correspond to constrained versions of the respective combinatorial problems which are often $\mathcal{NP}$-hard to solve. In this context, however, the constraints have some specific (binary) structure which may allow polynomial time solution methods . Further research should focus on other specific problem classes like shortest paths or knapsack problems to see whether a polynomial time solution is also possible in other cases.

Furthermore, the complexity of the overall algorithm can possibly be improved by implementing "warm start recursions", i.e. by using the solution of a previous iteration of Algorithm 3 as a "warm start solution" for the next ε-constraint problem to solve. Since only a few values of the involved cost coefficients change, the former optimal solution is either feasible for the new problem, or it could at least provide a good bound on the optimal objective value of the new subproblem that has to be solved.

Alternatively, neighborhood search techniques (based on a simple exchange of items) could be used to find efficient solutions for the associated multiple objective problem with binary objectives that correspond to optimal solutions of Problem (B_ν). However, if such an approach is applied, it must be guaranteed that the set of efficient solutions is connected with respect to the considered swap operation, i.e. all efficient

solutions can be constructed by simple swaps involving efficient solutions only. Note that this property fails for many classes of combinatorial problems in general (cf. [25]). However, due to the simple structure of the involved binary objectives, connectedness may hold for the efficient set in this context (cf. [26]).

Acknowledgment. This work was partially supported by the German Academic Exchange Service DAAD (PPP Portugal).

References

[1] J. Gorski, S. Ruzika, On k-max Optimization, Operations Research Letters 37 (1) (2009) 23–26.
[2] V. Chankong, Y. Y. Haimes, Multiobjective Decision Making: Theory and Methodology, Elsevier Science Publishing, New York, 1983.
[3] J. Edmonds, D. R. Fulkerson, Bottleneck extrema, Journal of Combinatorial Theory 8 (1970) 299–306.
[4] M. R. Garey, D. S. Johnson, Computers and Intractability, Freeman, New York, 1979.
[5] C. Jorgensen, S. Powell, Solving $0-1$ Minimax Problems, Journal of the Operational Research Society 38 (6) (1987) 515–522.
[6] J. Krarup, P. M. Pruzan, Reducibility of minimax to minisum 0-1 programming problems, European Journal of Operations Research 6 (1981) 125–132.
[7] I. I. Melamed, P. I. K. Sigal, An investigation of linear convolution of criteria in multicriteria discrete programming, Computational Mathematics and Mathematical Physics 35 (8) (1995) 1009–1017.
[8] I. I. Melamed, P. I. K. Sigal, The linear convolution of criteria in the bicriteria traveling salesman problem, Computational Mathematics and Mathematical Physics 37 (8) (1997) 902–905.
[9] I. I. Melamed, P. I. K. Sigal, Numerical Analysis of tricriteria tree and assignment problems, Computational Mathematics and Mathematical Physics 38 (10) (1998) 1707–1714.
[10] I. I. Melamed, P. I. K. Sigal, N. Y. Vladimirova, Study of linear parametrization of criteria in the bicriteria knapsack problem, Computational Mathematics and Mathematical Physics 39 (5) (1999) 721–726.
[11] P. Hansen, Bicriterion path problems, in: G. Fandel, T. Gal (Eds.), Multicriteria decision making: theory and applications, vol. 177 of *Lecture Notes in Economics and Mathematical Systems*, Springer, Heidelberg, 109–127, 1980.
[12] E. Q. V. Martins, On a multicriteria shortest path problem, European Journal of Operational Research 16 (1984) 236–245.
[13] O. Berman, D. Einav, G. Handler, The constrained bottleneck problem in network, Operations Research 38 (1990) 178–181.
[14] B. Pelegrin, P. Fernandez, On the sum-max bicriterion path problem, Computers and Operations Research 25 (12) (1998) 1043–1054.
[15] L. Pinto, C. Bornstein, N. Maculan, The tricriterion shortest path problem with at least two bottleneck objectives, European Journal of Operational Research 198 (2009) 387–391.
[16] L. Pinto, M. Pascoal, On algorithms for the tricriteria shortest path problem with two bottleneck objective functions, Computers & Operations Research 37 (2010) 1774–1779.
[17] M. Minoux, Solving Combinatorial Problems with combined MIN-MAX-MIN-SUM objective and applications, Mathematical Programming 45 (1989) 361–372.
[18] A. P. Punnen, K. P. K. Nair, An $\mathcal{O}(m \log n)$ algorithm for the max + sum spanning tree problem, European Journal of Operational Research 89 (1996) 423–426.
[19] R. E. Burkard, F. Rendl, Lexicographic bottleneck problems, Operations Research Letters 10 (1991) 303–308.
[20] H. I. Calvete, P. M. Mateo, Lexicographic optimization in generalized network flow problems, Journal of the Operational Research Society 49 (1998) 519–529.
[21] F. D. Croce, V. T. Paschos, A. Tsoukiàs, An improved general procedure for lexicographic bottleneck problems, Operations Research Letters 24 (4) (1999) 187–194.
[22] A. P. Punnen, Y. P. Aneja, Lexicographic balanced optimization problems, Operations Research Letters 32 (2004) 27–30.
[23] V. Aggarwal, Y. P. Aneja, K. P. K. Nair, Minimal spanning tree subject to a side constraint, Computers & Operations Research 9 (4) (1982) 287–296.
[24] H. N. Gabow, R. E. Tarjan, Efficient Algorithms for a Family of Matroid Intersection Problems, Journal of Algorithms 5 (1984) 80–131.
[25] J. Gorski, K. Klamroth, S. Ruzika, Connectedness of Efficient Solutions in Multiple Objective Combinatorial Optimization, Tech. Rep. 310, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institut für Angewandte Mathematik, 2006.
[26] J. Gorski, L. Paquete, F. Pedrosa, Greedy algorithms for a class of knapsack problems with binary weights, Tech. Rep. BUW-AMNA-OPAP 09/02, Bergische Universität Wuppertal, Fachbereich Mathematik und Naturwissenschaften, 2009.