



# The Power of the Weighted Sum Scalarization for Approximating Multiobjective Optimization Problems

Cristina Bazgan<sup>1</sup> · Stefan Ruzika<sup>2</sup> · Clemens Thielen<sup>3</sup>  · Daniel Vanderpooten<sup>1</sup>

Accepted: 24 October 2021 / Published online: 22 November 2021

© The Author(s) 2021

## Abstract

We determine the power of the weighted sum scalarization with respect to the computation of approximations for general multiobjective minimization and maximization problems. Additionally, we introduce a new multi-factor notion of approximation that is specifically tailored to the multiobjective case and its inherent trade-offs between different objectives. For minimization problems, we provide an efficient algorithm that computes an approximation of a multiobjective problem by using an exact or approximate algorithm for its weighted sum scalarization. In case that an exact algorithm for the weighted sum scalarization is used, this algorithm comes arbitrarily close to the best approximation quality that is obtainable by supported solutions – both with respect to the common notion of approximation and with respect to the new multi-factor notion. Moreover, the algorithm yields the currently best approximation results for several well-known multiobjective minimization problems. For maximization problems, however, we show that a polynomial approximation guarantee can, in general, not be obtained in more than one of the objective functions simultaneously by supported solutions.

**Keywords** Approximation algorithms · Multiobjective optimization · Weighted sum scalarization

---

This work was supported by the DFG grants RU 1524/6-1 and TH 1852/4-1 as well as by a bilateral cooperation project funded by the German Academic Exchange Service (DAAD, Project-ID 57388848) and by Campus France, PHC PROCOPE 2018 (Project no. 40407WF).

✉ Clemens Thielen  
clemens.thielen@tum.de

Extended author information available on the last page of the article.

## 1 Introduction

Almost any real-world optimization problem asks for optimizing more than one objective function (e.g., the minimization of cost and time in transportation systems or the maximization of profit and safety in investments). Clearly, these objectives are conflicting, often incommensurable, and, yet, they have to be taken into account simultaneously. The discipline dealing with such problems is called *multiobjective optimization*. Typically, multiobjective optimization problems are solved according to the Pareto principle of optimality: a solution is called *efficient* (or *Pareto optimal*) if no other feasible solution exists that is not worse in any objective function and better in at least one objective. The images of the efficient solutions in the objective space are called *nondominated points*. In contrast to single objective optimization, where one typically asks for one optimal solution, the main goal of multiobjective optimization is to compute the set of all nondominated points and, for each of them, one corresponding efficient solution. Each of these solutions corresponds to a different compromise among the set of objectives and may potentially be relevant for a decision maker.

Several results in the literature, however, show that multiobjective optimization problems are hard to solve exactly [11, 12] and, in addition, the cardinalities of the set of nondominated points (the *nondominated set*) and the set of efficient solutions (the *efficient set*) may be exponentially large for discrete problems (and are typically infinite for continuous problems). This impairs the applicability of exact solution methods to real-life problems and provides a strong motivation for studying *approximations of multiobjective optimization problems*.

Both exact and approximate solution methods for multiobjective optimization problems often resort to using single objective auxiliary problems, which are called *scalarizations* of the original multiobjective problem. This refers to the transformation of a multiobjective optimization problem into a single objective auxiliary problem based on a procedure that might use additional parameters, auxiliary points, or variables. The resulting scalarized optimization problems are then solved using methods from single objective optimization and the obtained solutions are interpreted in the context of Pareto optimality.

The simplest and most widely used scalarization technique is the *weighted sum scalarization* (see, e.g., [12]). Here, the scalarized auxiliary problem is constructed by assigning a weight to each of the objective functions and summing up the resulting weighted objective functions in order to obtain the objective function of the scalarized problem. If the weights are chosen to be positive, then every optimal solution of the resulting *weighted sum problem* is efficient. Moreover, the weighted sum scalarization does not change the feasible set and, in many cases, boils down to the single objective version of the given multiobjective problem — which represents an important advantage of this scalarization especially for combinatorial problems. However, only some efficient solutions (called *supported solutions*) can be obtained by means of the weighted sum scalarization, while many other efficient solutions (called *unsupported solutions*) cannot. Consequently, a natural question is to determine which approximations of the whole efficient set can be obtained by using this very important scalarization technique.

## 1.1 Previous Work

Besides many specialized approximation algorithms for particular multiobjective optimization problems, there exist several general approximation methods that can be applied to broad classes of multiobjective problems. An extensive survey of these general approximation methods is provided in [18].

Most general approximation methods for multiobjective problems are based on the seminal work of Papadimitriou and Yannakakis [21], who present a method for generating a  $(1 + \varepsilon, \dots, 1 + \varepsilon)$ -approximation for general multiobjective minimization and maximization problems with a constant number of positive-valued, polynomially computable objective functions. They show that a  $(1 + \varepsilon, \dots, 1 + \varepsilon)$ -approximation with size polynomial in the encoding length of the input and  $\frac{1}{\varepsilon}$  always exists. Moreover, their results show that the construction of such an approximation is possible in (fully) polynomial time, i.e., the problem admits a *multiobjective (fully) polynomial-time approximation scheme* or *MPTAS (MFPTAS)*, if and only if a certain auxiliary problem called the *gap problem* can be solved in (fully) polynomial time.

More recent articles building upon the results of [21] present methods that additionally yield bounds on the size of the computed  $(1 + \varepsilon, \dots, 1 + \varepsilon)$ -approximation relative to the size of the smallest  $(1 + \varepsilon, \dots, 1 + \varepsilon)$ -approximation possible [4, 10, 22]. Moreover, it has recently been shown in [5] that an even better  $(1, 1 + \varepsilon, \dots, 1 + \varepsilon)$ -approximation (i.e., an approximation that is exact in one objective function and  $(1 + \varepsilon)$ -approximate in all other objective functions) always exists, and that such an approximation can be computed in (fully) polynomial time if and only if the so-called *dual restrict problem* (introduced in [10]) can be solved in (fully) polynomial time.

Other works study how the weighted sum scalarization can be used in order to compute a set of solutions such that the convex hull of their images in the objective space yields an approximation guarantee of  $(1 + \varepsilon, \dots, 1 + \varepsilon)$  [7–9]. Using techniques similar to ours, Diakonikolas and Yannakakis [9] show that such a so-called  $\varepsilon$ -convex *Pareto set* can be computed in (fully) polynomial time if and only if the weighted sum scalarization admits a (fully) polynomial-time approximation scheme. Additionally, they consider questions regarding the cardinality of  $\varepsilon$ -convex Pareto sets.

Besides the general approximation methods mentioned above that work for both minimization and maximization problems, there exist several general approximation methods that are restricted either to minimization problems or to maximization problems.

For *minimization* problems, there are two general approximation methods that are both based on using (approximations of) the weighted sum scalarization. The previously best general approximation method for multiobjective minimization problems with an arbitrary constant number of objectives that uses the weighted sum scalarization can be obtained by combining two results of Glaßer et al. [15, 16]. They introduce another auxiliary problem called the *approximate domination problem*, which is similar to the gap problem. Glaßer et al. show that, if this problem is solvable in polynomial time for some approximation factor  $\alpha \geq 1$ , then an approximating set providing an approximation factor of  $\alpha \cdot (1 + \varepsilon)$  in every objective function can be computed in fully polynomial time for every  $\varepsilon > 0$ . Moreover, they show that the approximate domination problem with  $\alpha := \sigma \cdot p$  can be solved by using

a  $\sigma$ -approximation algorithm for the weighted sum scalarization of the  $p$ -objective problem. Together, this implies that a  $((1 + \varepsilon) \cdot \sigma \cdot p, \dots, (1 + \varepsilon) \cdot \sigma \cdot p)$ -approximation can be computed in fully polynomial time for  $p$ -objective minimization problems provided that the objective functions are positive-valued and polynomially computable and a  $\sigma$ -approximation algorithm for the weighted sum scalarization exists. As this result is not explicitly stated in [15, 16], no bounds on the running time are provided.

For *biobjective* minimization problems, Halffmann et al. [17] show how to obtain a  $(\sigma \cdot (1 + 2\varepsilon), \sigma \cdot (1 + \frac{2}{\varepsilon}))$ -approximation for any given  $0 < \varepsilon \leq 1$  if a polynomial-time  $\sigma$ -approximation algorithm for the weighted sum scalarization is given.

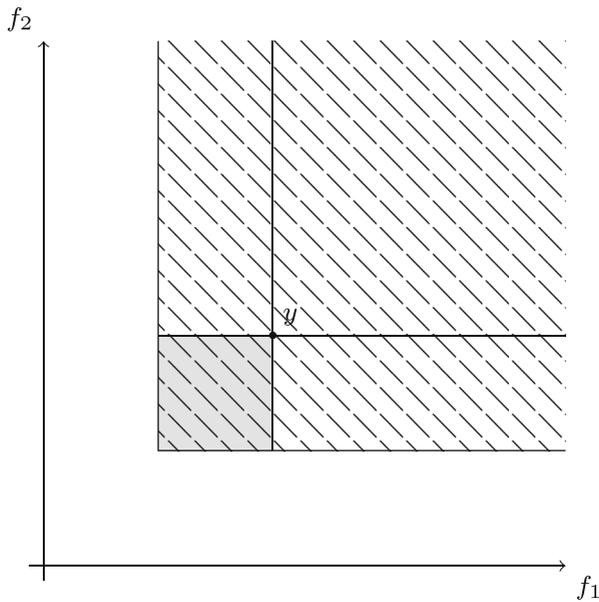
Obtaining general approximation methods for multiobjective *maximization* problems using the weighted sum scalarization seems to be much harder than for minimization problems. Indeed, Glaßer et al. [15] show that certain translations of approximability results from the weighted sum scalarization of an optimization problem to the multiobjective version that work for minimization problems are not possible in general for maximization problems.

An approximation method specifically designed for multiobjective maximization problems is presented by Bazgan et al. [3]. Their method is applicable to biobjective maximization problems that satisfy an additional structural assumption on the set of feasible solutions and the objective functions: For each two feasible solutions none of which approximates the other one by a factor of  $\alpha$  in both objective functions, a third solution approximating both given solutions in both objective functions by a certain factor depending on  $\alpha$  and a parameter  $c$  must be computable in polynomial time. The approximation factor obtained by the algorithm then depends on  $\alpha$  and  $c$ .

## 1.2 Our Contribution

Our contribution is twofold: First, in order to better capture the approximation quality in the context of multiobjective optimization problems, we introduce a new notion of approximation for the multiobjective case. This new notion comprises the common notion of approximation, but is specifically tailored to the multiobjective case and its inherent trade-offs between different objectives. Second, we provide a precise analysis of the approximation quality obtainable for multiobjective optimization problems by means of an exact or approximate algorithm for the weighted sum scalarization – with respect to both the common and the new notion of approximation.

In order to motivate the new notion of approximation, consider the biobjective case, in which a  $(2 + \varepsilon, 2 + \varepsilon)$ -approximation can be obtained from the results of Glaßer et al. [15, 16] using an exact algorithm for the weighted sum scalarization. As illustrated in Fig. 1, this approximation guarantee is actually too pessimistic: Since each point  $y$  in the image of the approximating set is nondominated (since it is the image of an optimal solution of the weighted sum scalarization), no images of feasible solutions can be contained in the shaded region. Thus, every feasible solution is actually either  $(1, 2 + \varepsilon)$ - or  $(2 + \varepsilon, 1)$ -approximated. Consequently, the approximation quality obtained in this case can be more accurately described by using *two vectors of approximation factors*. In order to capture such situations and allow for a more precise analysis of the approximation quality obtained for



**Fig. 1** Image space of a biobjective minimization problem. The point  $y$  in the image of the approximating set  $(2 + \varepsilon, 2 + \varepsilon)$ -approximates all points in the hatched region. If  $y$  is nondominated, no images of feasible solutions can be contained in the shaded region, so every image in the hatched region is actually either  $(1, 2 + \varepsilon)$ - or  $(2 + \varepsilon, 1)$ -approximated

multiobjective problems, our new *multi-factor notion of approximation* uses a *set of vectors of approximation factors* instead of only a single vector.

The second part of our contribution consists of a detailed analysis of the approximation quality obtainable by using the weighted sum scalarization – both for multiobjective minimization problems and for multiobjective maximization problems. For minimization problems, we provide an efficient algorithm that approximates a multiobjective problem using an exact or approximate algorithm for its weighted sum scalarization. We analyze the approximation quality obtained by the algorithm both with respect to the common notion of approximation that uses only a single vector of approximation factors as well as with respect to the new multi-factor notion. With respect to the common notion, our algorithm matches the best previously known approximation guarantee of  $(\sigma \cdot p + \varepsilon, \dots, \sigma \cdot p + \varepsilon)$  obtainable for  $p$ -objective minimization problems and any  $\varepsilon > 0$  from a  $\sigma$ -approximation algorithm for the weighted sum scalarization. More importantly, we show that this result is best-possible in the sense that it comes arbitrarily close to the best approximation guarantee obtainable by supported solutions for the case that an exact algorithm is used to solve the weighted sum problem (i.e., when  $\sigma = 1$ ).

When analyzing the algorithm with respect to the new multi-factor notion of approximation, however, a much stronger approximation result is obtained. Here, we show that every feasible solution is approximated with some (possibly different) vector  $(\alpha_1, \dots, \alpha_p)$  of approximations factors such that  $\sum_{j:\alpha_j > 1} \alpha_j = \sigma \cdot p + \varepsilon$ .

In particular, the worst-case approximation factor of  $\sigma \cdot p + \varepsilon$  can actually be tight *in at most one objective* for any feasible point. This shows the multi-factor notion of approximation yields a much stronger approximation result by allowing a refined analysis of the obtained approximation guarantee. Moreover, for  $\sigma = 1$ , we show that the obtained multi-factor approximation result comes arbitrarily close to the best multi-factor approximation result obtainable by supported solutions. We also demonstrate that our algorithm applies to a large variety of multiobjective minimization problems and yields the currently best approximation results for several problems.

Multiobjective maximization problems, however, turn out to be much harder to approximate by using the weighted sum scalarization. Here, we show that a polynomial approximation guarantee can, in general, not be obtained in more than one of the objective functions simultaneously when using only supported solutions.

In summary, our results yield essentially tight bounds on the power of the weighted sum scalarization with respect to the approximation of multiobjective minimization and maximization problems – both in the common notion of approximation and in the new multi-factor notion.

The remainder of the paper is organized as follows: In Section 2, we formally introduce multiobjective optimization problems and provide the necessary definitions concerning their approximation. Section 3 contains our general approximation algorithm for minimization problems (Section 3.1). Moreover, we show in Section 3.2 that the obtained approximation results are tight. Section 4 concludes the paper by listing applications of our general approximation algorithm and directions for future work. In the appendix, we present both a faster approximation algorithm for minimization problems in the biobjective case (Appendix A) as well as our impossibility result for maximization problems (Appendix B).

## 2 Preliminaries

In the following, we consider a general multiobjective minimization or maximization problem  $\Pi$  of the following form (where either all objective functions are to be minimized or all objective functions are to be maximized):

$$\begin{array}{ll} \min / \max & f(x) = (f_1(x), \dots, f_p(x)) \\ \text{s. t.} & x \in X \end{array}$$

Here, as usual, we assume a constant number  $p \geq 2$  of objectives. The elements  $x \in X$  are called *feasible solutions* and the set  $X$ , which is assumed to be nonempty, is referred to as the *feasible set*. An image  $y = f(x)$  of a feasible solution  $x \in X$  is also called a (*feasible*) *point*. We let  $Y := f(X) := \{f(x) : x \in X\} \subseteq \mathbb{R}^p$  denote the *set of feasible points*.

We assume that the objective functions take only positive rational values and are polynomially computable. Moreover, for each  $j \in \{1, \dots, p\}$ , we assume that there exist strictly positive rational lower and upper bounds  $\text{LB}(j)$ ,  $\text{UB}(j)$  of polynomial encoding length such that  $\text{LB}(j) \leq f_j(x) \leq \text{UB}(j)$  for all  $x \in X$ . We let  $\text{LB} := \min_{j=1, \dots, p} \text{LB}(j)$  and  $\text{UB} := \max_{j=1, \dots, p} \text{UB}(j)$ .

**Definition 1** For a minimization problem  $\Pi$ , we say that a point  $y = f(x) \in Y$  is *dominated* by another point  $y' = f(x') \in Y$  if  $y' \neq y$  and

$$y'_j = f_j(x') \leq f_j(x) = y_j \text{ for all } j \in \{1, \dots, p\}.$$

Similarly, for a maximization problem  $\Pi$ , we say that a point  $y = f(x) \in Y$  is *dominated* by another point  $y' = f(x') \in Y$  if  $y' \neq y$  and

$$y'_j = f_j(x') \geq f_j(x) = y_j \text{ for all } j \in \{1, \dots, p\}.$$

If the point  $y = f(x)$  is not dominated by any other point  $y'$ , we call  $y$  *nondominated* and the feasible solution  $x \in X$  *efficient*. The set  $Y_N$  of nondominated points is called the *nondominated set* and the set  $X_E$  of efficient solutions is called the *efficient set* or *Pareto set*.

## 2.1 Notions of Approximation

We first recall the standard definitions of approximation for single objective optimization problems.

**Definition 2** Consider a single objective optimization problem  $\Pi$  and let  $\alpha \geq 1$ . If  $\Pi$  is a minimization problem, we say that a feasible solution  $x \in X$   $\alpha$ -*approximates* another feasible solution  $x' \in X$  if  $f(x) \leq \alpha \cdot f(x')$ . If  $\Pi$  is a maximization problem, we say that a feasible solution  $x \in X$   $\alpha$ -*approximates* another feasible solution  $x' \in X$  if  $\alpha \cdot f(x) \geq f(x')$ . A feasible solution that  $\alpha$ -approximates every feasible solution of  $\Pi$  is called an  $\alpha$ -*approximation* for  $\Pi$ .

A (*polynomial-time*)  $\alpha$ -*approximation algorithm* is an algorithm that, for every instance  $I$  with encoding length  $|I|$ , computes an  $\alpha$ -approximation for  $\Pi$  in time bounded by a polynomial in  $|I|$ .

The following definition extends the concept of approximation to the multiobjective case.

**Definition 3** Let  $\alpha = (\alpha_1, \dots, \alpha_p) \in \mathbb{R}^p$  with  $\alpha_j \geq 1$  for all  $j \in \{1, \dots, p\}$ .

For a minimization problem  $\Pi$ , we say that a feasible solution  $x \in X$   $\alpha$ -*approximates* another feasible solution  $x' \in X$  (or, equivalently, that the feasible point  $y = f(x)$   $\alpha$ -*approximates* the feasible point  $y' = f(x')$ ) if

$$f_j(x) \leq \alpha_j \cdot f_j(x') \text{ for all } j \in \{1, \dots, p\}.$$

Similarly, for a maximization problem  $\Pi$ , we say that a feasible solution  $x \in X$   $\alpha$ -*approximates* another feasible solution  $x' \in X$  (or, equivalently, that the feasible point  $y = f(x)$   $\alpha$ -*approximates* the feasible point  $y' = f(x')$ ) if

$$\alpha_j \cdot f_j(x) \geq f_j(x') \text{ for all } j \in \{1, \dots, p\}.$$

The standard notion of approximation for multiobjective optimization problems used in the literature is the following one.

**Definition 4** Let  $\alpha = (\alpha_1, \dots, \alpha_p) \in \mathbb{R}^p$  with  $\alpha_j \geq 1$  for all  $j \in \{1, \dots, p\}$ .

A set  $P \subseteq X$  of feasible solutions is called an  $\alpha$ -approximation for the multiobjective problem  $\Pi$  if, for any feasible solution  $x' \in X$ , there exists a solution  $x \in P$  that  $\alpha$ -approximates  $x'$ .

In the following definition, we generalize the standard notion of approximation for multiobjective problems by allowing a *set of vectors of approximation factors* instead of only a single vector, which allows for tighter approximation results.

**Definition 5** Let  $\mathcal{A} \subseteq \mathbb{R}^p$  be a set of vectors with  $\alpha_j \geq 1$  for all  $\alpha \in \mathcal{A}$  and all  $j \in \{1, \dots, p\}$ . Then a set  $P \subseteq X$  of feasible solutions is called a (*multi-factor*)  $\mathcal{A}$ -approximation for the multiobjective problem  $\Pi$  if, for any feasible solution  $x' \in X$ , there exist a solution  $x \in P$  and a vector  $\alpha \in \mathcal{A}$  such that  $x$   $\alpha$ -approximates  $x'$ .

Note that, in the case where  $\mathcal{A} = \{(\alpha_1, \dots, \alpha_p)\}$  is a singleton, an  $\mathcal{A}$ -approximation for a multiobjective problem according to Definition 5 is equivalent to an  $(\alpha_1, \dots, \alpha_p)$ -approximation according to Definition 4.

## 2.2 Weighted Sum Scalarization

Given a  $p$ -objective optimization problem  $\Pi$  and a vector  $w = (w_1, \dots, w_p) \in \mathbb{R}^p$  with  $w_j > 0$  for all  $j \in \{1, \dots, p\}$ , the *weighted sum problem* (or *weighted sum scalarization*)  $\Pi^{\text{WS}}(w)$  associated with  $\Pi$  is defined as the following single objective optimization problem:

$$\begin{aligned} \min / \max \quad & \sum_{j=1}^p w_j \cdot f_j(x) \\ \text{s. t.} \quad & x \in X \end{aligned}$$

**Definition 6** A feasible solution  $x \in X$  is called *supported* if there exists a vector  $w = (w_1, \dots, w_p) \in \mathbb{R}^p$  of positive weights such that  $x$  is an optimal solution of the weighted sum problem  $\Pi^{\text{WS}}(w)$ . In this case, the feasible point  $y = f(x) \in Y$  is called a *supported point*. The set of all supported solutions is denoted by  $X_S$ .

It is well-known that every supported point is nondominated and, correspondingly, every supported solution is efficient (cf. [12]).

In the following, we assume that there exists a polynomial-time  $\sigma$ -approximation algorithm  $\text{WS}_\sigma$  for the weighted sum problem, where  $\sigma \geq 1$  can be either a constant or a function of the input size. When calling  $\text{WS}_\sigma$  with some specific weight vector  $w$ , we denote this by  $\text{WS}_\sigma(w)$ . This algorithm then returns a solution  $\hat{x}$  such that  $\sum_{j=1}^p w_j f_j(\hat{x}) \leq \sigma \cdot \sum_{j=1}^p w_j f_j(x)$  for all  $x \in X$ , if  $\Pi$  is a minimization problem, and  $\sigma \cdot \sum_{j=1}^p w_j f_j(\hat{x}) \geq \sum_{j=1}^p w_j f_j(x)$  for all  $x \in X$ , if  $\Pi$  is a maximization problem. The running time of algorithm  $\text{WS}_\sigma$  is denoted by  $T_{\text{WS}_\sigma}$ .

The following result shows that a  $\sigma$ -approximation for the weighted sum problem is also a  $\sigma$ -approximation of any solution in at least one of the objectives.

**Lemma 1** *Let  $\hat{x} \in X$  be a  $\sigma$ -approximation for  $\Pi^{\text{WS}}(w)$  for some positive weight vector  $w \in \mathbb{R}^p$ . Then, for any feasible solution  $x \in X$ , there exists at least one  $i \in \{1, \dots, p\}$  such that  $\hat{x}$   $\sigma$ -approximates  $x$  in objective  $f_i$ .*

*Proof* Consider the case where  $\Pi$  is a multiobjective *minimization* problem (the proof for the case where  $\Pi$  is a maximization problem works analogously). Then, we must show that, for any feasible solution  $x \in X$ , there exists at least one  $i \in \{1, \dots, p\}$  such that  $f_i(\hat{x}) \leq \sigma \cdot f_i(x)$ .

Assume by contradiction that there exists some  $x' \in X$  such that  $f_j(\hat{x}) > \sigma \cdot f_j(x')$  for all  $j \in \{1, \dots, p\}$ . Then, we obtain  $\sum_{j=1}^p w_j \cdot f_j(\hat{x}) > \sigma \cdot \sum_{j=1}^p w_j \cdot f_j(x')$ , which contradicts the assumption that  $\hat{x}$  is a  $\sigma$ -approximation for  $\Pi^{\text{WS}}(w)$ .  $\square$

### 3 A Multi-Factor Approximation Result for Minimization Problems

In this section, we study the approximation of multiobjective *minimization* problems by (approximately) solving weighted sum problems. In Section 3.1, we propose a multi-factor approximation algorithm that significantly improves upon the  $((1 + \varepsilon) \cdot \sigma \cdot p, \dots, (1 + \varepsilon) \cdot \sigma \cdot p)$ -approximation algorithm that can be derived from Glaßer et al. [15]. Moreover, we show in Section 3.2 that the resulting approximation is tight.

#### 3.1 Approximation Results

**Proposition 1** *Let  $\bar{x} \in X$  be a feasible solution of  $\Pi$  and let  $b = (b_1, \dots, b_p)$  be such that  $b_j \leq f_j(\bar{x}) \leq (1 + \varepsilon) \cdot b_j$  for  $j = 1, \dots, p$  and some  $\varepsilon > 0$ . Applying  $\text{WS}_\sigma(w)$  with  $w_j := \frac{1}{b_j}$  for  $j = 1, \dots, p$  yields a solution  $\hat{x}$  that  $(\alpha_1, \dots, \alpha_p)$ -approximates  $\bar{x}$  for some  $\alpha_1, \dots, \alpha_p \geq 1$  such that  $\alpha_i \leq \sigma$  for at least one  $i \in \{1, \dots, p\}$  and*

$$\sum_{j:\alpha_j > 1} \alpha_j = (1 + \varepsilon) \cdot \sigma \cdot p.$$

*Proof* Since  $\hat{x}$  is the solution returned by  $\text{WS}_\sigma(w)$ , we have

$$\sum_{j=1}^p \frac{1}{b_j} f_j(\hat{x}) \leq \sigma \cdot \left( \sum_{j=1}^p \frac{1}{b_j} f_j(\bar{x}) \right) \leq \sigma \cdot (1 + \varepsilon) \cdot \left( \sum_{j=1}^p 1 \right) = (1 + \varepsilon) \cdot \sigma \cdot p.$$

Since  $\frac{1}{b_j} \geq \frac{1}{f_j(\bar{x})}$ , we get  $\sum_{j=1}^p \frac{f_j(\hat{x})}{f_j(\bar{x})} \leq \sum_{j=1}^p \frac{1}{b_j} f_j(\hat{x})$ , which yields

$$\sum_{j=1}^p \frac{f_j(\hat{x})}{f_j(\bar{x})} \leq (1 + \varepsilon) \cdot \sigma \cdot p.$$

Setting  $\alpha_j := \max \left\{ 1, \frac{f_j(\hat{x})}{f_j(\bar{x})} \right\}$  for  $j = 1, \dots, p$ , we have

$$\sum_{j:\alpha_j > 1} \alpha_j \leq (1 + \varepsilon) \cdot \sigma \cdot p.$$

The worst case approximation factors  $\alpha_j$  are then obtained when equality holds in the previous inequality.

Moreover, by Lemma 1, there exists at least one  $i \in \{1, \dots, p\}$  such that  $f_i(\hat{x}) \leq \sigma \cdot f_i(\bar{x})$ . Thus, we have  $\alpha_i \leq \sigma$  for at least one  $i \in \{1, \dots, p\}$ , which proves the claim.  $\square$

Proposition 1 motivates to apply the given  $\sigma$ -approximation algorithm  $WS_\sigma$  for  $\Pi^{WS}$  iteratively for different weight vectors  $w$  in order to obtain an approximation of the multiobjective minimization problem  $\Pi$ . This is formalized in Algorithm 1, whose correctness and running time are established in Theorem 1.

---

**Algorithm 1** An  $\mathcal{A}$ -approximation for  $p$ -objective minimization problems.

---

```

input : an instance of a  $p$ -objective minimization problem  $\Pi$ ,  $\varepsilon > 0$ , a
          $\sigma$ -approximation algorithm  $WS_\sigma$  for the weighted sum problem
output: an  $\mathcal{A}$ -approximation  $P$  for problem  $\Pi$ 

1  $P \leftarrow \emptyset$ 
2  $\varepsilon' \leftarrow \frac{\varepsilon}{\sigma \cdot p}$ 
3 for  $j \leftarrow 1$  to  $p$  do
4    $u_j \leftarrow$  largest integer such that  $LB(j) \cdot (1 + \varepsilon')^{u_j} \leq UB(j)$ 
5 for  $k \leftarrow 1$  to  $p$  do
6    $i_k \leftarrow 0$ 
7   foreach  $(i_1, \dots, i_p)$  such that  $i_\ell \in \{1, \dots, u_\ell\}$  for  $\ell < k$ , and
    $i_\ell \in \{0, \dots, u_\ell\}$  for  $\ell > k$  do
8     for  $j \leftarrow 1$  to  $p$  do
9        $b_j \leftarrow LB(j) \cdot (1 + \varepsilon')^{i_j}$ 
10       $w_j \leftarrow \frac{1}{b_j}$ 
11       $x \leftarrow WS_\sigma(w)$ 
12       $P \leftarrow P \cup \{x\}$ 
13 return  $P$ 

```

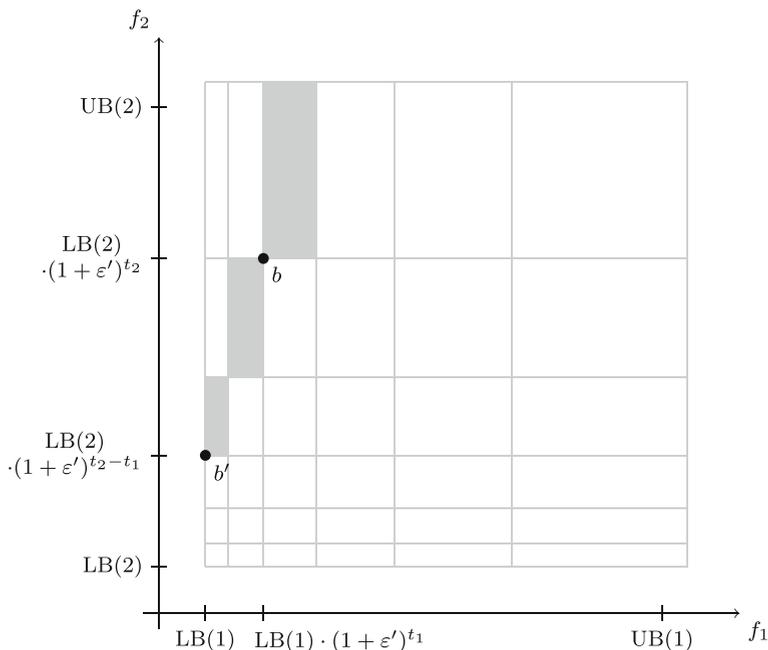
---

**Theorem 1** For a  $p$ -objective minimization problem, Algorithm 1 outputs an  $\mathcal{A}$ -approximation where

$$\mathcal{A} = \left\{ (\alpha_1, \dots, \alpha_p) : \alpha_1, \dots, \alpha_p \geq 1, \alpha_i \leq \sigma \text{ for at least one } i, \text{ and } \sum_{j:\alpha_j > 1} \alpha_j = \sigma \cdot p + \varepsilon \right\}$$

$$\text{in time } T_{WS_\sigma} \cdot \sum_{i=1}^p \prod_{j \neq i} \left\lceil \log_{1+\frac{\varepsilon}{\sigma p}} \frac{UB(j)}{LB(j)} \right\rceil \in \mathcal{O} \left( T_{WS_\sigma} \left( \frac{\sigma}{\varepsilon} \log \frac{UB}{LB} \right)^{p-1} \right).$$

*Proof* In order to approximate all feasible solutions, we can iteratively apply Proposition 1 with  $\varepsilon' := \frac{\varepsilon}{\sigma \cdot p}$  instead of  $\varepsilon$ , leading to the modified constraint on the sum of the  $\alpha_j$  where the right-hand side becomes  $(1 + \varepsilon') \cdot \sigma \cdot p = \sigma \cdot p + \varepsilon$ . More precisely, we iterate with  $b_j = LB(j) \cdot (1 + \varepsilon')^{i_j}$  and  $i_j = 0, \dots, u_j$ , where  $u_j$  is the largest integer such that  $LB(j) \cdot (1 + \varepsilon')^{u_j} \leq UB(j)$ , for each  $j \in \{1, \dots, p\}$ . Actually, this iterative application of Proposition 1 involves redundant weight vectors. More precisely, consider a weight vector  $w = (w_1, \dots, w_p)$  where  $w_j = \frac{1}{b_j}$  with  $b_j = LB(j) \cdot (1 + \varepsilon')^{t_j}$  for  $j = 1, \dots, p$ , and let  $k$  be an index such that  $t_k = \min_{j=1, \dots, p} t_j$ . Then problem  $\Pi^{WS}(w)$  is equivalent to problem  $\Pi^{WS}(w')$  with  $w'_j = \frac{1}{b'_j}$ , where  $b'_j = LB(j) \cdot (1 + \varepsilon')^{t_j - t_k}$  for  $j = 1, \dots, p$ . Therefore, it is sufficient to consider all weight vectors  $w$  for which at least one component  $w_k$  is set to  $\frac{1}{LB(k)}$  (see Fig. 2 for an illustration). The running time follows.  $\square$



**Fig. 2** Weight vectors and subdivision of the objective space in Algorithm 1. The weight vector  $w = (\frac{1}{b_1}, \dots, \frac{1}{b_p})$  with  $b_j = LB(j) \cdot (1 + \varepsilon')^{t_j}$  for  $j = 1, \dots, p$  is equivalent to the weight vector  $w' = (\frac{1}{b'_1}, \dots, \frac{1}{b'_p})$  obtained by reducing all exponents  $t_j$  by their minimum. The solution  $WS_\sigma(w')$  returned for  $w'$  is then used to approximate all solutions with images in the shaded (hyper-) rectangles

Note that the set  $\mathcal{A}$  in Theorem 1 is infinite. By rounding up each approximation factor  $\alpha_j$  to the nearest power of  $1 + \varepsilon$ , however, it is possible to obtain the same result for the *finite* set

$$\bar{\mathcal{A}} = \left\{ (\bar{\alpha}_1, \dots, \bar{\alpha}_p) : \bar{\alpha}_j = (1 + \varepsilon)^{l_j} \text{ for some } l_j \in \mathbb{Z}_{\geq 0} \ (j = 1, \dots, p), \right. \\ \left. \bar{\alpha}_i \leq (1 + \varepsilon) \cdot \sigma \text{ for at least one } i, \text{ and } \sum_{j:\bar{\alpha}_j > 1} \bar{\alpha}_j = (1 + \varepsilon) \cdot (\sigma \cdot p + \varepsilon) \right\}.$$

Also note that, depending on the structure of the weighted sum algorithm  $WS_\sigma$ , the practical running time of Algorithm 1 could be improved by not solving every weighted sum problem from scratch, but using the information obtained in previous iterations.

Moreover, as illustrated in Fig. 2, Algorithm 1 also directly yields a subdivision of the objective space into hyperrectangles such that all solutions whose images are in the same hyperrectangle are approximated by the same solution (possibly with different approximation guarantees): For each weight vector  $w = (\frac{1}{b_1}, \dots, \frac{1}{b_p})$  considered in the algorithm (where  $b_k = LB(k)$  for at least one  $k$ ), all solutions  $\bar{x}$  with images in the hyperrectangles  $\times_{j=1}^p [b_j \cdot (1 + \varepsilon)^\ell, b_j \cdot (1 + \varepsilon)^{\ell+1}]$  for  $\ell = 0, 1, \dots$  are approximated by the solution returned by  $WS_\sigma(w)$ .

When the weighted sum problem can be solved exactly in polynomial time, Theorem 1 immediately yields the following result:

**Corollary 1** *If  $WS_\sigma = WS_1$  is an exact algorithm for the weighted sum problem, Algorithm 1 outputs an  $\mathcal{A}$ -approximation where*

$$\mathcal{A} = \left\{ (\alpha_1, \dots, \alpha_p) : \alpha_1, \dots, \alpha_p \geq 1, \alpha_i = 1 \text{ for at least one } i, \text{ and } \sum_{j:\alpha_j > 1} \alpha_j = p + \varepsilon \right\}$$

in time  $\mathcal{O}\left(T_{WS_1} \left(\frac{1}{\varepsilon} \log \frac{UB}{LB}\right)^{p-1}\right)$ .

Another special case worth mentioning is the situation where the weighted sum problem admits a polynomial-time approximation scheme. Here, similar to the case in which an exact algorithm is available for the weighted sum problem, we can still obtain a set of vectors  $\alpha$  of approximation factors with  $\sum_{j:\alpha_j > 1} \alpha_j = p + \varepsilon$  while only losing the property that at least one  $\alpha_i$  equals 1.

**Corollary 2** *If the weighted sum problem admits a polynomial-time  $(1 + \tau)$ -approximation for any  $\tau > 0$ , then, for any  $\varepsilon > 0$  and any  $0 < \tau < \frac{\varepsilon}{p}$ , Algorithm 1 can be used to compute an  $\mathcal{A}$ -approximation where*

$$\mathcal{A} = \left\{ (\alpha_1, \dots, \alpha_p) : \alpha_1, \dots, \alpha_p \geq 1, \alpha_i \leq 1 + \tau \text{ for at least one } i, \text{ and } \sum_{j:\alpha_j > 1} \alpha_j = p + \varepsilon \right\}$$

in time  $\mathcal{O}\left(T_{WS_{1+\tau}}\left(\frac{1+\tau}{\varepsilon-\tau \cdot p} \log \frac{UB}{LB}\right)^{p-1}\right)$ .

*Proof* Given  $\varepsilon > 0$  and  $0 < \tau < \frac{\varepsilon}{p}$ , apply Algorithm 1 with  $\varepsilon - \tau \cdot p$  and  $\sigma := 1 + \tau$ . □

Since any component of a vector in the set  $\mathcal{A}$  from Theorem 1 can get arbitrarily close to  $\sigma \cdot p + \varepsilon$  in the worst case, the best “classical” approximation result using only a single vector of approximation factors that is obtainable from Theorem 1 reads as follows:

**Corollary 3** *Algorithm 1 computes a  $(\sigma \cdot p + \varepsilon, \dots, \sigma \cdot p + \varepsilon)$ -approximation in time  $\mathcal{O}\left(T_{WS_\sigma}\left(\frac{1}{\varepsilon} \log \frac{UB}{LB}\right)^{p-1}\right)$ .*

### 3.2 Tightness Results

When solving the weighted sum problem exactly, Corollary 1 states that Algorithm 1 obtains a set  $\mathcal{A}$  of approximation factors in which  $\sum_{j:\alpha_j > 1} \alpha_j = p + \varepsilon$  for each  $\alpha = (\alpha_1, \dots, \alpha_p) \in \mathcal{A}$ .

The following theorem shows that this multi-factor approximation result is arbitrarily close to the best possible result obtainable by supported solutions:

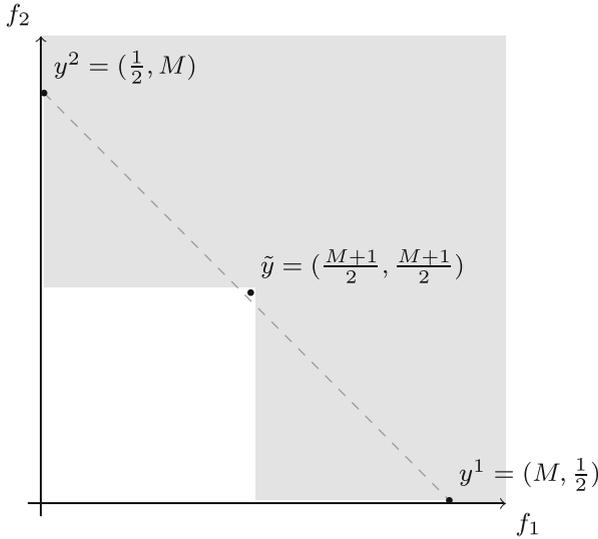
**Theorem 2** *For  $\varepsilon > 0$ , let*

$$\mathcal{A} := \{\alpha \in \mathbb{R}^p : \alpha_1, \dots, \alpha_p \geq 1, \alpha_i = 1 \text{ for at least one } i, \text{ and } \sum_{j:\alpha_j > 1} \alpha_j = p - \varepsilon\}.$$

*Then there exists an instance of a  $p$ -objective minimization problem for which the set  $X_S$  of supported solutions is not an  $\mathcal{A}$ -approximation.*

*Proof* In the following, we only specify the set  $Y$  of images. A corresponding instance consisting of a set  $X$  of feasible solutions and an objective function  $f$  can then easily be obtained, e. g., by setting  $X := Y$  and  $f := \text{id}_{\mathbb{R}^p}$ .

For  $M > 0$ , let  $Y := \{y^1, \dots, y^p, \tilde{y}\}$  with  $y^1 = \left(M, \frac{1}{p}, \dots, \frac{1}{p}\right)$ ,  $y^2 = \left(\frac{1}{p}, M, \frac{1}{p}, \dots, \frac{1}{p}\right), \dots, y^p = \left(\frac{1}{p}, \dots, \frac{1}{p}, M\right)$  and  $\tilde{y} = \left(\frac{M+1}{p}, \dots, \frac{M+1}{p}\right)$ . Note that



**Fig. 3** Image space of the instance constructed in the proof of Theorem 2 for  $p = 2$ . The shaded region is  $\{(1, 2 - \varepsilon), (2 - \varepsilon, 1)\}$ -approximated by the supported points  $y^1, y^2$

the point  $\tilde{y}$  is unsupported, while  $y^1, \dots, y^p$  are supported (an illustration for the case  $p = 2$  is provided in Fig. 3).

Moreover, the ratio of the  $j$ -th components of the points  $y^j$  and  $\tilde{y}$  is exactly

$$\frac{M}{(M+1)/p} = p \cdot \frac{M}{M + 1},$$

which is larger than  $p - \varepsilon$  for  $M > \frac{p}{\varepsilon} - 1$ . Consequently, for such  $M$ , the point  $\tilde{y}$  is not  $\alpha$ -approximated by any of the supported points  $y^1, \dots, y^p$  for any  $\alpha \in \mathcal{A}$ , which proves the claim.  $\square$

We remark that the set of points  $Y$  constructed in the proof of Theorem 2 can easily be obtained from instances of many well-known multiobjective minimization problems such as multiobjective shortest path, multiobjective spanning tree, multiobjective minimum ( $s$ - $t$ -) cut, or multiobjective TSP (e.g., for multiobjective shortest path, a collection of  $p + 1$  disjoint  $s$ - $t$ -paths whose cost vectors correspond to the points  $y^1, \dots, y^p, \tilde{y}$  suffices). Consequently, the result from Theorem 2 holds for each of these specific problems as well.

Moreover, note that also the classical approximation result obtained in Corollary 3 is arbitrarily close to best possible in case that the weighted sum problem is solved exactly: While Corollary 3 shows that a  $(p + \varepsilon, \dots, p + \varepsilon)$ -approximation is obtained from Algorithm 1 when solving the weighted sum problem exactly, the instance constructed in the proof of Theorem 2 shows that the supported solutions do not yield

an approximation guarantee of  $(p - \varepsilon, \dots, p - \varepsilon)$  for any  $\varepsilon > 0$ . This yields the following theorem:

**Theorem 3** *For any  $\varepsilon > 0$ , there exists an instance of a  $p$ -objective minimization problem for which the set  $X_S$  of supported solutions is not a  $(p - \varepsilon, \dots, p - \varepsilon)$ -approximation.*

## 4 Conclusion

The weighted sum scalarization is the most frequently used method to transform a multiobjective into a single objective optimization problem. In this article, we contribute to a better understanding of the quality of approximations for general multiobjective optimization problems which rely on this scalarization technique. To this end, we refine and extend the common notion of approximation quality in multiobjective optimization. As we show, the resulting multi-factor notion of approximation more accurately describes the approximation quality in multiobjective contexts.

For multiobjective *minimization* problems, we also present an efficient approximation algorithm, which turns out to be best possible under some additional assumptions. This algorithm can be applied to a large variety of minimization problems and improves upon the previously best approximation results for several well-known problems. If the weighted sum scalarization admits a polynomial-time approximation scheme, our algorithm yields a multi-factor approximation where each feasible solution is approximated with some approximation guarantee  $(\alpha_1, \dots, \alpha_p)$  such that  $\sum_{j:\alpha_j>1} \alpha_j = p + \varepsilon$ . This yields the best known approximation results for the multiobjective versions of the weighted planar TSP [20] and minimum weight planar vertex cover [1]. If the weighted sum scalarization admits a polynomial-time  $\sigma$ -approximation algorithm, our algorithm yields both a (classical)  $(\sigma \cdot p + \varepsilon, \dots, \sigma \cdot p + \varepsilon)$ -approximation and a multi-factor approximation where each feasible solution is approximated with some approximation guarantee  $(\alpha_1, \dots, \alpha_p)$  such that  $\sum_{j:\alpha_j>1} \alpha_j = \sigma \cdot p + \varepsilon$  and  $\alpha_i \leq \sigma$  for at least one  $i$ . These results yield the best known approximation guarantees for many well-studied problems whose single objective version does not admit a polynomial time approximation scheme unless  $P = NP$  (and, consequently, the multiobjective version does not admit an MPTAS under the same assumption). Problems of this kind include, e.g., minimum weight vertex cover [2], minimum  $k$ -spanning tree [14], minimum weight edge dominating set [13], and minimum metric  $k$ -center [19], whose single objective versions admit 2-approximation algorithms, as well as the minimum weight set cover problem, where a  $(1 + \ln |S|)$ -approximation exists with  $|S|$  denoting the cardinality of the ground set to cover [6].

Interestingly, it is easy to show that similar approximation results cannot be obtained for multiobjective *maximization* problems (see Appendix B).

Since the new multi-factor notion of approximation is independent of the specific algorithms used here, a natural direction for future research is to analyze new and existing approximation algorithms more precisely with the help of this new notion. This may yield both a better understanding of existing approaches as well as more accurate approximation results.

## Appendix A: Biobjective Minimization Problems

We now focus on *biobjective* minimization problems. We first specialize some of the general results of Section 3.1 to the case  $p = 2$ . Afterwards, we propose a specific approximation algorithm for biobjective problems, which significantly improves upon the running time of Algorithm 1 in the case where an exact algorithm  $WS_1$  for the weighted sum problem is available.

Theorem 1, which is the main general result of Section 3.1, can trivially be specialized to the case  $p = 2$ . It is more interesting to consider the situation where the weighted sum can be solved exactly, corresponding to Corollary 1. In this case, we obtain the following result:

**Corollary 4** *If  $WS_\sigma = WS_1$  is an exact algorithm for the weighted sum problem and  $p = 2$ , Algorithm 1 yields an  $\mathcal{A}$ -approximation where*

$$\mathcal{A} = \{(1, 2 + \varepsilon), (2 + \varepsilon, 1)\}$$

in time  $\mathcal{O}\left(T_{WS_1} \frac{1}{\varepsilon} \log \frac{UB}{LB}\right)$ .

It is worth pointing out that, unlike in Corollary 1, the set  $\mathcal{A}$  of approximation factors is now *finite*. This type of result can be interpreted as a *disjunctive* approximation result: Algorithm 1 outputs a set  $P$  ensuring that, for any  $x \in X$ , there exists  $x' \in P$  such that  $x'$   $(1, 2 + \varepsilon)$ -approximates  $x$  or  $x'$   $(2 + \varepsilon, 1)$ -approximates  $x$ .

In the biobjective case, we may scale the weights in the weighted sum problem to be of the form  $(\gamma, 1)$  for some  $\gamma > 0$ . In the following, we make use of this observation and refer to a weight vector  $(\gamma, 1)$  simply as  $\gamma$ .

Algorithm 2 is a refinement of Algorithm 1 in the biobjective case when an exact algorithm  $WS_1$  for the weighted sum problem is available. Algorithm 1 requires to test all the  $u_1 + u_2 + 1$  weights  $\left(\frac{1}{LB(1)}, \frac{1}{LB(2)(1+\varepsilon)^{u_2}}\right), \left(\frac{1}{LB(1)}, \frac{1}{LB(2)(1+\varepsilon)^{u_2-1}}\right), \dots, \left(\frac{1}{LB(1)}, \frac{1}{LB(2)}\right), \left(\frac{1}{LB(1)(1+\varepsilon)}\right), \left(\frac{1}{LB(1)(1+\varepsilon)^2}\right), \dots, \left(\frac{1}{LB(1)(1+\varepsilon)^{u_1}}, \frac{1}{LB(2)}\right)$ , or equivalently the  $u_1 + u_2 + 1$  weights of the form  $(\gamma_t, 1)$ , where  $\gamma_t = \frac{LB(2)}{LB(1)}(1 + \varepsilon)^{u_2 - t + 1}$  for  $t = 1, \dots, u_1 + u_2 + 1$ . Instead of testing all these weights, Algorithm 2 considers only a subset of these weights. More precisely, in each iteration, the algorithm selects a subset of consecutive weights  $\{\gamma_\ell, \dots, \gamma_r\}$ , solves  $WS_1(\gamma_t)$  for the weight  $\gamma_t$  with  $t = \lfloor \frac{\ell+r}{2} \rfloor$ , and decides whether 0, 1, or 2 of the subsets  $\{\gamma_\ell, \dots, \gamma_t\}$  and  $\{\gamma_t, \dots, \gamma_r\}$  need to be investigated further. This process can be viewed as developing a binary tree where the root, which corresponds to the initialization, requires solving two weighted sum problems, while each other node requires solving one

weighted sum problem. This representation is useful to bound the running time of our algorithm. The following technical result on binary trees will be useful for this purpose:

**Lemma 2** *A binary tree with height  $h$  and  $k$  nodes with two children contains  $\mathcal{O}(k \cdot h)$  nodes.*

*Proof* In order to show the claimed upper bound on the number of nodes, we first show that any binary tree  $T$  with height  $h$  and  $k$  nodes with two children that has the maximum possible number of nodes among all such binary trees must have the following property: If  $v$  is a node with two children at level  $\ell$ , then all nodes  $u$  at the levels  $0, \dots, \ell - 1$  must also have two children.

So assume by contradiction that  $T$  is a binary tree maximizing the number of nodes among all trees with height  $h$  and  $k$  nodes with two children, but  $T$  does not have this property. Then there exists a node  $v$  in  $T$  with two children at some level  $\ell$  and a node  $u$  with at most one child at a lower level  $\ell' \in \{0, \dots, \ell - 1\}$ . Then, the binary tree  $T'$ , that results from making one node  $w$  that is a child of  $v$  in  $T$  an (additional) child of  $u$ , also has height  $h$ , contains  $k$  nodes with two children, and has the same number of nodes as  $T$ . Moreover, the level of  $w$  in  $T'$  changes to  $\ell' + 1 < \ell + 1$ . Hence, also the level of any leaf of the subtree rooted at  $w$  must have decreased by at least one. Thus, giving any leaf of this subtree an additional child in  $T'$  would yield a binary tree of height  $h$  and  $k$  nodes with two children, and a strictly larger number of nodes than  $T$ , contradicting the maximality of  $T$ .

By the above property, in any binary tree maximizing the number of nodes among the trees satisfying the assumptions of the lemma, there are only nodes with two children on all levels  $i < h' := \lfloor \log_2(k + 1) \rfloor$  and only nodes with at most one child on all levels  $i > h'$ . Level  $h'$  may contain nodes with two children, but there is at least one node with only one child on this level.

Consequently, there are at most  $k$  nodes in total on the levels  $0, \dots, h' - 1$  and at most  $k + 1$  nodes at level  $h'$ . Moreover, there are at most  $2(k + 1)$  nodes at level  $h' + 1$ , each of which is the root of a subtree (path) consisting of at most  $h - h'$  nodes (each with at most one child). Overall, this proves an upper bound of at most  $k + (k + 1) + 2(k + 1) \cdot (h - h') \in \mathcal{O}(k \cdot h)$  on the number of nodes in the tree.  $\square$

**Theorem 4** *For a biobjective minimization problem, Algorithm 2 returns a  $\{(1, 2 + \varepsilon), (2 + \varepsilon, 1)\}$ -approximation in time*

$$\mathcal{O} \left( T_{\text{WS}_1} \cdot \log \left( \frac{1}{\varepsilon} \cdot \log \frac{\text{UB}}{\text{LB}} \right) \cdot \log \frac{\text{UB}}{\text{LB}} \right).$$

*Proof* The approximation guarantee of Algorithm 2 derives from Theorem 1. We just need to prove that the subset of weights used here is sufficient to preserve the approximation guarantee.

In lines 21–22, the weights  $\gamma_i$  for  $i = \ell + 1, \dots, t - 1$  are not considered if  $x^\ell$   $(1, 2 + \varepsilon)$ -approximates  $x^t$  or if  $x^t$   $(2 + \varepsilon, 1)$ -approximates  $x^\ell$ . We show that, indeed, these

weights are not needed. To this end, first observe that any solution  $x^i := \text{WS}_1(\gamma_i)$  for  $i \in \{\ell + 1, \dots, t - 1\}$  is such that

$$f_1(x^\ell) \leq f_1(x^i) \leq f_1(x^t) \quad \text{and} \\ f_2(x^\ell) \geq f_2(x^i) \geq f_2(x^t).$$

since  $\gamma_\ell > \gamma_i > \gamma_t$ . Thus, if  $x^\ell$   $(1, 2 + \varepsilon)$ -approximates  $x^t$ , we obtain

$$f_2(x^\ell) \leq (2 + \varepsilon) \cdot f_2(x^t) \leq (2 + \varepsilon) \cdot f_2(x^i),$$

---

**Algorithm 2** A  $\{(1, 2 + \varepsilon), (2 + \varepsilon, 1)\}$ -approximation for biobjective minimization problems.

---

```

input : an instance of a biobjective minimization problem  $\Pi$ ,  $\varepsilon > 0$ , an exact
        algorithm  $\text{WS}_1$  for the weighted sum problem
output: a  $\{(1, 2 + \varepsilon), (2 + \varepsilon, 1)\}$ -approximation  $P$  for problem  $\Pi$ 
1   $\varepsilon' \leftarrow \frac{\varepsilon}{p}$ 
2   $u_1 \leftarrow$  largest integer such that  $LB(1)(1 + \varepsilon')^{u_1} \leq UB(1)$ 
3   $u_2 \leftarrow$  largest integer such that  $LB(2)(1 + \varepsilon')^{u_2} \leq UB(2)$ 
4   $\gamma_1 \leftarrow \frac{LB(2)}{LB(1)}(1 + \varepsilon')^{u_2}$ ;  $\gamma_{u_1+u_2+1} \leftarrow \frac{LB(2)}{LB(1)}(1 + \varepsilon')^{-u_1}$ 
5   $x^1 \leftarrow \text{WS}_1(\gamma_1)$ ;  $x^{u_1+u_2+1} \leftarrow \text{WS}_1(\gamma_{u_1+u_2+1})$ 
6   $Q \leftarrow \emptyset$ 
7  if  $x^1$   $(1, 2 + \varepsilon)$ -approximates  $x^{u_1+u_2+1}$  then  $P \leftarrow \{x^1\}$ ;
8  else
9  | if  $x^{u_1+u_2+1}$   $(2 + \varepsilon, 1)$ -approximates  $x^1$  then  $P \leftarrow \{x^{u_1+u_2+1}\}$ ;
10 | else
11 | |  $P \leftarrow \{x^1, x^{u_1+u_2+1}\}$ 
12 | |  $Q \leftarrow \{(1, u_1 + u_2 + 1)\}$ 
13 while  $Q \neq \emptyset$  do
14 | Select  $(\ell, r)$  from  $Q$ 
15 |  $Q \leftarrow Q \setminus \{(\ell, r)\}$ 
16 |  $t \leftarrow \lfloor \frac{\ell+r}{2} \rfloor$ 
17 |  $\gamma_t = \frac{LB(2)}{LB(1)}(1 + \varepsilon')^{u_2-t+1}$ 
18 |  $x^t \leftarrow \text{WS}_1(\gamma_t)$ 
19 | if  $x^\ell$  does not  $(1, 2 + \varepsilon)$ -approximate  $x^t$  or  $x^r$  does not
    |  $(2 + \varepsilon, 1)$ -approximate  $x^t$  then
20 | |  $P \leftarrow P \cup \{x^t\}$ 
21 | | if  $t \geq \ell + 2$  and  $x^\ell$  does not  $(1, 2 + \varepsilon)$ -approximate  $x^t$  and  $x^t$  does not
    | |  $(2 + \varepsilon, 1)$ -approximate  $x^\ell$  then
22 | | |  $Q \leftarrow Q \cup \{(\ell, t)\}$ 
23 | | if  $t \leq r - 2$  and  $x^t$  does not  $(1, 2 + \varepsilon)$ -approximate  $x^r$  and  $x^r$  does not
    | |  $(2 + \varepsilon, 1)$ -approximate  $x^t$  then
24 | | |  $Q \leftarrow Q \cup \{(t, r)\}$ 
25 return  $P$ 

```

---

which shows that  $x^\ell$  also  $(1, 2 + \varepsilon)$ -approximates  $x^i$ . Therefore,  $x^i$  and the corresponding weight  $\gamma_i$  are not needed.

Similarly, if  $x^t$   $(2 + \varepsilon, 1)$ -approximates  $x^\ell$ , we have

$$f_1(x^t) \leq (2 + \varepsilon) \cdot f_1(x^\ell) \leq (2 + \varepsilon) \cdot f_1(x^i),$$

which shows that  $x^t$   $(2 + \varepsilon, 1)$ -approximates  $x^i$ . Therefore  $x^i$  and the corresponding weight  $\gamma_i$  are again not needed.

In lines 23–24, the weights  $\gamma_i$  for  $i = t + 1, \dots, r - 1$  are not considered if  $x^t$   $(1, 2 + \varepsilon)$ -approximates  $x^r$  or if  $x^r$   $(2 + \varepsilon, 1)$ -approximates  $x^t$  for similar reasons.

Also, in line 19,  $x^t$  can be discarded and the weights  $\gamma_i$  for  $i = \ell + 1, \dots, r - 1$  can be ignored if  $x^\ell$   $(1, 2 + \varepsilon)$ -approximates  $x^t$  and  $x^r$   $(2 + \varepsilon, 1)$ -approximates  $x^t$ . Indeed, using similar arguments as before, we obtain that  $x^\ell$   $(1, 2 + \varepsilon)$ -approximates  $x^i$  for  $i = \ell + 1, \dots, t$  and  $x^r$   $(2 + \varepsilon, 1)$ -approximates  $x^i$  for  $i = t, \dots, r - 1$  in this case. Consequently, compared to Algorithm 1, only superfluous weights are discarded in Algorithm 2 and the approximation guarantee follows by Theorem 1.

We now prove the claimed bound on the running time. Algorithm 2 explores a set of weights of cardinality  $u_1 + u_2 + 1 = \left\lfloor \log_{1+\varepsilon'} \frac{UB(1)}{LB(1)} \right\rfloor + \left\lfloor \log_{1+\varepsilon'} \frac{UB(2)}{LB(2)} \right\rfloor + 1$ . The running time is obtained by bounding the number of calls to algorithm  $WS_1$ , which corresponds to the number of nodes of the binary tree implicitly developed by the algorithm. The height of this tree is  $\log_2(u_1 + u_2 + 1) \in \mathcal{O} \left( \log \left( \frac{1}{\varepsilon} \cdot \log \frac{UB}{LB} \right) \right)$ .

In order to bound the number of nodes with two children in the tree, we observe that we generate such a node (i.e. add the pairs  $(\ell, t)$  and  $(t, r)$  to  $\mathcal{Q}$ ) only if  $x^\ell$  does not  $(1, 2 + \varepsilon)$ -approximate  $x^t$  and  $x^t$  does not  $(2 + \varepsilon, 1)$ -approximate  $x^\ell$ , and also  $x^t$  does not  $(1, 2 + \varepsilon)$ -approximate  $x^r$  and  $x^r$  does not  $(2 + \varepsilon, 1)$ -approximate  $x^t$ . Hence, whenever a node with two children is generated, the corresponding solution  $x^t$  does neither  $(1, 2 + \varepsilon)$  nor  $(2 + \varepsilon, 1)$ -approximate any previously generated solution and vice versa, so their objective values in both of the two objective functions must differ by more than a factor  $(2 + \varepsilon)$ . Using that the  $j$ th objective value of any feasible solution is between  $LB(j)$  and  $UB(j)$ , this implies that there can be at most

$$\min \left\{ \log_{2+\varepsilon} \left( \frac{UB(1)}{LB(1)} \right); \log_{2+\varepsilon} \left( \frac{UB(2)}{LB(2)} \right) \right\} \in \mathcal{O} \left( \log \frac{UB}{LB} \right)$$

nodes with two children in the tree.

Using the obtained bounds on the height of the tree and the number of nodes with two children, Lemma 2 shows that the total number of nodes in the tree is

$$\mathcal{O} \left( \log \left( \frac{1}{\varepsilon} \cdot \log \frac{UB}{LB} \right) \cdot \log \frac{UB}{LB} \right),$$

which proves the claimed bound on the running time. □

### Appendix B: Maximization Problems

We now demonstrate that the weighted sum scalarization is much less powerful for approximating multiobjective *maximization* problems. Indeed, the following theorem

shows that, for maximization problems, a polynomial approximation factor can, in general, not be obtained in more than one of the objective functions simultaneously even if the approximating set consists of all the supported solutions:

**Theorem 5** *For any  $p \geq 2$  and any polynomial  $\text{pol}$ , there exists an instance  $I$  of a  $p$ -objective maximization problem such that at least one unsupported solution is not approximated with an approximation guarantee of  $2^{\text{pol}(|I|)}$  in  $p - 1$  of the objective functions by any supported solution.*

*Proof* Given  $p \geq 2$  and a polynomial  $\text{pol}$ , consider the  $p$ -objective maximization problem where each instance  $I$  is given by a  $(p+1)$ -tuple  $(x^1, \dots, x^p, \tilde{x})$  of pairwise different vectors  $x^1, \dots, x^p, \tilde{x} \in \mathbb{Z}^p$  and the feasible set is  $X = \{x^1, \dots, x^p, \tilde{x}\}$ .

Given the encoding length  $|I|$  of such an instance, we set  $M := 2^{\text{pol}(|I|)} + 1$  and  $f(x^1) = (M, \frac{1}{p}, \dots, \frac{1}{p})$ ,  $f(x^2) = (\frac{1}{p}, M, \frac{1}{p}, \dots, \frac{1}{p})$ ,  $\dots$ ,  $f(x^p) = (\frac{1}{p}, \dots, \frac{1}{p}, M)$  and  $f(\tilde{x}) = (\frac{M}{p}, \dots, \frac{M}{p})$ . Then, the solution  $\tilde{x}$  is unsupported, while  $x^1, \dots, x^p$  are supported.

Moreover, the ratio of the  $j$ -th components of the images  $f(\tilde{x})$  and  $f(x^\ell)$  for any  $j \neq \ell$  is exactly  $M = 2^{\text{pol}(|I|)} + 1 > 2^{\text{pol}(|I|)}$ , which shows that  $x^\ell$  does not yield an approximation guarantee of  $2^{\text{pol}(|I|)}$  in objective function  $f_j$  for any  $j \neq \ell$ .  $\square$

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Baker, B.S.: Approximation algorithms for NP-complete problems on planar graphs. *J. ACM* **41**(1), 153–180 (1994)
2. Bar-Yehuda, R., Even, S.: A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms* **2**(2), 198–203 (1981)
3. Bazgan, C., Gourvès, L., Monnot, J.: Approximation with a fixed number of solutions of some multiobjective maximization problems. *Journal of Discrete Algorithms* **22**, 19–29 (2013)
4. Bazgan, C., Jamain, F., Vanderpooten, D.: Approximate Pareto sets of minimal size for multi-objective optimization problems. *Oper. Res. Lett.* **43**(1), 1–6 (2015)
5. Bazgan, C., Herzel, A., Ruzika, S., Thielen, C., Vanderpooten, D.: One-exact approximate Pareto sets. *J. Glob. Optim.* **80**, 87–115 (2021)
6. Chvátal, V.: A greedy heuristic for the set-covering problem. *Math. Oper. Res.* **4**(3), 233–235 (1979)
7. Daskalakis, C., Diakonikolas, I., Yannakakis, M.: How good is the chord algorithm? *SIAM J. Comput.* **45**(3), 811–858 (2016)
8. Diakonikolas, I.: Approximation of Multiobjective Optimization Problems. PhD thesis, Columbia University (2011)

9. Diakonikolas, I., Yannakakis, M.: Succinct approximate convex Pareto curves. In: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 74–83 (2008)
10. Diakonikolas, I., Yannakakis, M.: Small approximate Pareto sets for biobjective shortest paths and other problems. *SIAM J. Comput.* **39**(4), 1340–1371 (2009)
11. Ehrgott, M.: Hard to say it's easy - four reasons why combinatorial multiobjective programmes are hard. In: Proceedings of the 14th International Conference on Multiple Criteria Decision Making (MCDM), Lecture Notes in Economics and Mathematical Systems, vol. 487, pp. 69–80 (1998)
12. Ehrgott, M.: *Multicriteria Optimization*. Springer, Berlin (2005)
13. Fujito, T., Nagamochi, H.: A 2-approximation algorithm for the minimum weight edge dominating set problem. *Discret. Appl. Math.* **118**(3), 199–207 (2002)
14. Garg, N.: Saving an epsilon: A 2-approximation for the k-MST problem in graphs. In: Proceedings of the 37th ACM Symposium on the Theory of Computing (STOC), pp. 396–402 (2005)
15. Glaßer, C., Reitwießner, C., Schmitz, H., Witek, M.: Approximability and hardness in multi-objective optimization. In: Proceedings of the 6th Conference on Computability in Europe (CiE), LNCS, vol. 6158, pp. 180–189 (2010)
16. Glaßer, C., Reitwießner, C., Schmitz, H., Witek, M.: Hardness and approximability in multi-objective optimization. Technical Report TR10-031, Electronic Colloquium on Computational Complexity (ECCC) (2010)
17. Halffmann, P., Ruzika, S., Thielen, C., Willems, D.: A general approximation method for bicriteria minimization problems. *Theor. Comput. Sci.* **695**, 1–15 (2017)
18. Herzel, A., Ruzika, S., Thielen, C.: Approximation methods for multiobjective optimization problems: a survey. *INFORMS Journal on Computing* **33**(4), 1284–1299 (2021)
19. Hochbaum, D.S., Shmoys, D.B.: A unified approach to approximation algorithms for bottleneck problems. *J. ACM* **33**(3), 533–550 (1986)
20. Klein, P.: A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM J. Comput.* **37**(6), 1926–1952 (2008)
21. Papadimitriou, C., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. In: Proceedings of the 41st Annual IEEE Symposium on the Foundations of Computer Science (FOCS), pp. 86–92 (2000)
22. Vassilvitskii, S., Yannakakis, M.: Efficiently computing succinct trade-off curves. *Theor. Comput. Sci.* **348**(2-3), 334–356 (2005)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

Cristina Bazgan<sup>1</sup> · Stefan Ruzika<sup>2</sup> · Clemens Thielen<sup>3</sup>  · Daniel Vanderpooten<sup>1</sup>

Cristina Bazgan  
bazgan@lamsade.dauphine.fr

Stefan Ruzika  
ruzika@mathematik.uni-kl.de

Daniel Vanderpooten  
daniel.vanderpooten@lamsade.dauphine.fr

<sup>1</sup> Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, 75016, Paris, France

<sup>2</sup> Department of Mathematics, University of Kaiserslautern, Paul-Ehrlich-Str. 14, 67663, Kaiserslautern, Germany

<sup>3</sup> TUM Campus Straubing, Technical University of Munich, Am Essigberg 3, 94315, Straubing, Germany