Journal of Global Optimization (2021) 80:87–115
https://doi.org/10.1007/s10898-020-00951-7

One-exact approximate Pareto sets

Arne Herzel[1,3] · Cristina Bazgan[2] · Stefan Ruzika[1] · Clemens Thielen[3] · Daniel Vanderpooten[2]

Received: 7 May 2019 / Accepted: 20 September 2020 / Published online: 18 November 2020
© The Author(s) 2020

Abstract
Papadimitriou and Yannakakis (Proceedings of the 41st annual IEEE symposium on the Foundations of Computer Science (FOCS), pp 86–92, 2000) show that the polynomial-time solvability of a certain auxiliary problem determines the class of multiobjective optimization problems that admit a polynomial-time computable $(1+\varepsilon, \ldots, 1+\varepsilon)$-approximate Pareto set (also called an *ε-Pareto set*). Similarly, in this article, we characterize the class of multiobjective optimization problems having a polynomial-time computable approximate ε-Pareto set that is *exact in one objective* by the efficient solvability of an appropriate auxiliary problem. This class includes important problems such as multiobjective shortest path and spanning tree, and the approximation guarantee we provide is, in general, best possible. Furthermore, for biobjective optimization problems from this class, we provide an algorithm that computes a one-exact ε-Pareto set of cardinality at most twice the cardinality of a smallest such set and show that this factor of 2 is best possible. For three or more objective functions, however, we prove that no constant-factor approximation on the cardinality of the set can be obtained efficiently.

Keywords Multiobjective optimization · Approximation algorithm · Approximate Pareto set · scalarization

Mathematics Subject Classification 90C29 · 90C59

1 Introduction

In many cases, real-world optimization problems involve several conflicting objectives, e.g., the minimization of cost and time in transportation systems or the maximization of profit and security in investments. In this context, solutions optimizing all objectives simultaneously

This work was supported by the bilateral cooperation project "Approximation methods for multiobjective optimization problems" funded by the German Academic Exchange Service (DAAD, Project-ID 57388848) and by Campus France, PHC PROCOPE 2018 (Project No. 40407WF) as well as by the DFG Grants RU 1524/6-1 and TH 1852/4-1.

✉ Arne Herzel
herzel@mathematik.uni-kl.de

Extended author information available on the last page of the article

usually do not exist. Therefore, in order to support decision making, so-called *efficient* (or *Pareto optimal*) solutions achieving a good compromise among the objectives are considered. More formally, a solution is said to be efficient if any other solution that is better in some objective is necessarily worse in at least one other objective. The image of an efficient solution in the objective space is called a *nondominated point*.

When no prior preference information is available, one main goal of multiobjective optimization is to determine the set of all nondominated points and provide, for each of them, one corresponding efficient solution.

Several results in the literature, however, show that multiobjective optimization problems are hard to solve exactly [4,5,8] 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 multiobjective optimization problems (and are typically infinite for continuous problems).

This impairs the applicability of exact solution methods to real-life optimization problems and provides a strong motivation for studying *approximations of multiobjective optimization problems*.

1.1 Related work

The systematic study of generally applicable approximation methods for multiobjective optimization problems started with the seminal work by Papadimitriou and Yannakakis [14]. They show that, for any $\varepsilon > 0$, any multiobjective optimization problem with a constant number of positive-valued, polynomial-time computable objective functions admits a $(1+\varepsilon, \ldots, 1+\varepsilon)$-approximate Pareto set (also called an *ε-Pareto set*) with cardinality polynomial in the encoding length of the input and $\frac{1}{\varepsilon}$. Moreover, they show that such a set is computable in polynomial time if and only if the following auxiliary problem called the *gap problem* ($\textsc{Gap}_\delta$) can be solved in polynomial time for a suitable value $\delta > 0$.[1]

Given an instance of a p-objective minimization problem and a vector $b \in \mathbb{R}^p$, either return a feasible solution x whose objective value $f(x) \in \mathbb{R}^p$ satisfies $f_j(x) \leq b_j$ for all j or answer correctly that there is no feasible solution x' with $f_j(x') \leq \frac{b_j}{1+\delta}$ for all j.

It should be noted that, even though the gap problem is often solvable for any $\delta > 0$, δ is not considered part of the input. Thus, an algorithm for $\textsc{Gap}_\delta$ is said to run in polynomial time if its running time is polynomial in the encoding lengths of the multiobjective instance and of the vector b, but its running time might depend on $\frac{1}{\delta}$ in a super-polynomial way. $\textsc{Gap}_\delta$ being solvable in *fully* polynomial time means that the time needed to solve an instance of $\textsc{Gap}_\delta$ is not only polynomial in the encoding lengths of the multiobjective instance and b, but also in $\frac{1}{\delta}$.[2]

The result by Papadimitriou and Yannakakis [14] shows that the (fully) polynomial-time solvability of $\textsc{Gap}_\delta$ provides a complete characterization of the class of multiobjective optimization problems for which ε-Pareto sets can be computed in (fully) polynomial time.

More recent articles building upon the results of [14] present methods that additionally yield bounds on the cardinality of the computed ε-Pareto set relative to the cardinality of a smallest ε-Pareto set possible [3,6,12,16].

[1] The definition of $\textsc{Gap}_\delta$ provided here is for minimization problems. The definition for maximization problems is completely analogous.

[2] We will use the term *fully polynomial* throughout this paper to state that some quantity is polynomial in the encoding length of the instance at hand and in the reciprocal of a certain accuracy parameter δ or ε.

Koltun and Papadimitriou [12] show that, if all feasible solutions of a biobjective optimization problem are given explicitly in the input (which is usually not the case for combinatorial problems, where the feasible set is in most cases given implicitly, and its cardinality is exponentially large in the input size), it is possible to compute an ε-Pareto set of minimum cardinality in polynomial time using a greedy procedure. This greedy procedure can be generalized to the case that the budget-constrained problem associated with the given biobjective optimization problem can be solved exactly in polynomial time [6]. For three or more objectives, however, computing a minimum-cardinality ε-Pareto set is NP-hard even if all feasible solutions are given explicitly.

Again for biobjective optimization problems, Vassilvitskii and Yannakakis [16] show that, using a polynomial-time algorithm for $\textsc{Gap}_\delta$ as a subroutine, it is possible to compute an ε-Pareto set whose cardinality is at most 3 times larger than the cardinality of a smallest ε-Pareto set in polynomial time. Moreover, this factor of 3 is shown to be best possible in two different ways: (1) No *generic* algorithm that uses only a routine for $\textsc{Gap}_\delta$ can obtain a factor smaller than 3 without solving $\textsc{Gap}_\delta$ for exponentially large values of $\frac{1}{\delta}$ (even if $\mathsf{P} = \mathsf{NP}$), and (2) for some biobjective optimization problems for which $\textsc{Gap}_\delta$ is polynomially solvable, it is NP-hard to obtain a factor smaller than 3. An alternative, simpler algorithm that also obtains a factor of 3 and is also usable for any problem for which a polynomial time algorithm for $\textsc{Gap}_\delta$ is available is presented in [3]. For three or more objectives, however, Vassilvitskii and Yannakakis [16] show that no generic algorithm based on solving $\textsc{Gap}_\delta$ can obtain any constant factor with respect to the cardinality of a smallest ε-Pareto set.

Diakonikolas and Yannakakis [6] show that, for a broad class of biobjective optimization problems including ShortestPath and SpanningTree, a factor of 2 can be obtained with respect to the cardinality of a smallest ε-Pareto set. To achieve this, they use subroutines for two different auxiliary problems called $\textsc{Restrict}_\delta$ and $\textsc{DualRestrict}_\delta$. Both of these problems are harder to solve than $\textsc{Gap}_\delta$ in the sense that any instance of $\textsc{Gap}_\delta$ can be solved by solving an instance of one of these problems, but there exist optimization problems (e.g., the biobjective knapsack problem [3]) for which $\textsc{Gap}_\delta$ can be solved in polynomial time but solving $\textsc{Restrict}_\delta$ or $\textsc{DualRestrict}_\delta$ is NP-hard. However, $\textsc{Restrict}_\delta$ and $\textsc{DualRestrict}_\delta$ are polynomially equivalent to each other for biobjective problems. The factor of 2 is again shown to be best possible in [6] in the sense that no generic algorithm based on $\textsc{Restrict}_\delta$ and $\textsc{DualRestrict}_\delta$ can obtain a smaller factor and, for some biobjective optimization problems for which $\textsc{Restrict}_\delta$ and $\textsc{DualRestrict}_\delta$ are polynomially solvable, it is NP-hard to obtain a smaller factor. For three or more objectives, however, it is not known whether $\textsc{Restrict}_\delta$ and $\textsc{DualRestrict}_\delta$ can be used to improve upon the results obtained via $\textsc{Gap}_\delta$ with respect to the computation of (small) ε-Pareto sets. Moreover, these two auxiliary problems are not polynomially equivalent anymore in the case of three or more objectives.

There are also many specialized approximation algorithms for particular multiobjective optimization problems available. Among those, there are two algorithms that actually yield approximations that are exact in one objective: For multiobjective ShortestPath, Tsaggouris and Zaroliagis [15] present a dynamic-programming-based algorithm that yields a $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-approximate Pareto set for any number of objective functions. For the min-cost-makespan scheduling problem, Angel et al. [1,2] present an algorithm computing a $(1, 1+\varepsilon)$-approximate Pareto set. Neither of these algorithms, however, is shown to yield a worst-case guarantee on the cardinality of the computed approximate Pareto set.

1.2 Our contribution

We consider general multiobjective optimization problems with an arbitrary, fixed number of objectives and show that, for any such problem, there exist polynomially-sized ε-Pareto sets that are *exact in one objective function*. Assuming without loss of generality that the first objective function is the one to be optimized exactly, we refer to such $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-approximate Pareto sets as *one-exact* ε-Pareto sets. Consequently, we improve upon the existence result for polynomially-sized ε-Pareto sets of Papadimitriou and Yannakakis [14] using the same assumptions. Moreover, the approximation guarantee of $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$ we provide is best possible in the sense that polynomially-sized approximate Pareto sets that are exact in more than one objective do, in general, not exist.

We then consider the class of multiobjective optimization problems for which the DualRestrict$_\delta$ subproblem considered in [6] can be solved in polynomial time. We first show that, for any constant number of objective functions, the polynomial-time solvability of DualRestrict$_\delta$ characterizes the class of optimization problems for which one-exact ε-Pareto sets can be computed in polynomial time. Consequently, even for more than two objective functions, our result provides a complete characterization of the approximation quality achievable for the class of optimization problems studied in [6], which includes ShortestPath, SpanningTree, and many more.

Moreover, we provide results about the cardinality of the computed one-exact ε-Pareto sets compared to the cardinality of a smallest such set. We show that the cardinality of a smallest one-exact ε-Pareto set (i.e., a one-exact ε-Pareto set having minimum cardinality among all one-exact ε-Pareto sets) can be much larger than the cardinality of a smallest ε-Pareto set (i.e., an ε-Pareto set having minimum cardinality among all ε-Pareto sets) even for biobjective optimization problems. We prove that, similar to the case of ε-Pareto sets, the cardinality of a smallest one-exact ε-Pareto set can be approximated up to a factor of 2 in the biobjective case by using a generic algorithm based on solving DualRestrict$_\delta$, and we show that this factor is best possible given our assumptions. For three or more objectives, however, it is again impossible to efficiently obtain any constant factor approximation on the cardinality by using only routines for DualRestrict$_\delta$.

For multiobjective SpanningTree, our generic algorithms yield the first polynomial-time methods for computing one-exact ε-Pareto sets when using the algorithm provided in [10] to solve DualRestrict$_\delta$. For multiobjective ShortestPath, using the algorithm from [11] to solve DualRestrict$_\delta$, our algorithms have running times competitive with the running time of the specialized algorithm for computing one-exact ε-Pareto sets for ShortestPath provided in [15]. This is particularly noteworthy since the algorithm from [15] is currently the algorithm with the best worst-case running time for computing ε-Pareto sets for ShortestPath even when no objective function is to be optimized exactly. Moreover, for the case of two objectives, our biobjective algorithm additionally provides a worst-case guarantee on the cardinality of the computed one-exact ε-Pareto set (while the algorithm from [15] provides no such guarantee).

2 Preliminaries

We consider general multiobjective minimization and maximization problems formally defined as follows:

Definition 1 (*Multiobjective Minimization/Maximization Problem*) A *multiobjective optimization problem* Π is given by a set of instances. Each instance I consists of a (finite or infinite) set X^I of feasible solutions and a vector $f^I = (f_1^I, \ldots, f_p^I)$ of p objective functions $f_i^I : X^I \to \mathbb{Q}$ for $i = 1 \ldots, p$. In a minimization problem, all objective functions f_i^I should be minimized, in a maximization problem, they should be maximized. The feasible set X^I might not be given explicitly.

Here, the number p of objective functions in a multiobjective optimization problem Π is assumed to be constant. The solutions of interest are those for which it is not possible to improve the value of one objective function without worsening the value of at least one other objective. Solutions with this property are called *efficient solutions*:

Definition 2 For an instance I of a minimization (maximization) problem, a solution $x \in X^I$ *dominates* another solution $x' \in X^I$ if $f_i^I(x) \leq f_i^I(x')$ ($f_i^I(x) \geq f_i^I(x')$) for $i = 1, \ldots, p$ and $f_i^I(x) < f_i^I(x')$ ($f_i^I(x) > f_i^I(x')$) for at least one i. A solution $x \in X^I$ is called *efficient* if it is not dominated by any other solution $x' \in X^I$. In this case, we call the corresponding image $f^I(x) \in f^I(X^I) \subseteq \mathbb{Q}^p$ a *nondominated point*. The set $X_E^I \subseteq X^I$ of all efficient solutions is called the *efficient set* (or *Pareto set*) and the set $Y_N^I := f^I(X_E^I)$ of nondominated points is called the *nondominated set*.

The goal in multiobjective optimization typically consists of computing the nondominated set Y_N^I and, for each nondominated point $y \in Y_N^I$, one corresponding efficient solution $x \in X_E^I$ with $f(x) = y$.

Throughout the paper, we make the standard assumption of rational, positive-valued, polynomial-time computable objective functions used in the context of approximation of multiobjective optimization problems (cf. [6,14,16]). Moreover, we make the following widely-used assumption (cf. [6,16]):

Assumption 1 For any multiobjective optimization problem Π, there exists a polynomial $\mathcal{P}$ such that, for any instance I of Π, there exists a polynomial-time computable value $M^I \leq \mathcal{P}(\text{enc}(I))$ such that $\text{enc}(f_i^I(x)) \leq M^I$ for any $x \in X^I$ and any $i \in \{1, \ldots, p\}$, where $\text{enc}(I)$ denotes the encoding length of the instance I and $\text{enc}(f_i^I(x))$ denotes the encoding length of the value $f_i^I(x) \in \mathbb{Q}_{>0}$ in binary. This, in particular, implies that, for any instance I and any objective function value $f_i^I(x)$, we have $2^{-M^I} \leq f_i^I(x) \leq 2^{M^I}$. Also, any two values $f_i^I(x)$ and $f_i^I(x')$ differ by at least 2^{-2M^I} if they are not equal.

In the following, we blur the distinction between the problem Π and a concrete instance $I = (X^I, f^I)$ and usually drop the superscript I indicating the dependence on the instance in X^I, f^I, M^I, etc.

Multiobjective optimization problems consist of objective functions that are to be minimized or objectives that are to be maximized (or even a combination of both). However, we only consider minimization objectives in this article. This is without loss of generality here since all our arguments can be straightforwardly adapted to maximization problems.

One of the main issues in multiobjective optimization problems is that the nondominated set often consists of exponentially many points, which renders the problem intractable (see, e.g., [7]). One way to overcome this obstacle is the concept of approximation. Instead of computing at least one corresponding efficient solution for each point in the nondominated set, we only require each image point in the objective space to be "almost" dominated by the image of a solution from the computed set.

Definition 3 (*Approximate Pareto set*) Let (X, f) be a multiobjective optimization problem and let $\alpha_i \geq 1$ for $i = 1, \ldots, p$. We say that a feasible solution $x \in X$ $(\alpha_1, \ldots, \alpha_p)$-*approximates* another feasible solution $x' \in X$ if $f(x)$ $(\alpha_1, \ldots, \alpha_p)$-*dominates* $f(x')$, i.e., for minimization problems, if $f_i(x) \leq \alpha_i \cdot f_i(x')$ and, for maximization problems, if $\alpha_i \cdot f_i(x) \geq f_i(x')$, for all $i = 1, \ldots, p$. A set $P_{(\alpha_1,\ldots,\alpha_p)} \subseteq X$ of feasible solutions is called an $(\alpha_1, \ldots, \alpha_p)$-*approximate Pareto set* if, for every feasible solution $x' \in X$, there exists a solution $x \in P_{(\alpha_1,\ldots,\alpha_p)}$ that $(\alpha_1, \ldots, \alpha_p)$-approximates x'. For $\varepsilon > 0$, a $(1+\varepsilon, \ldots, 1+\varepsilon)$-approximate Pareto set is called an ε-*Pareto set*.

Remark 1 If $\alpha_i = 1$ for two or more indices i, there exist optimization problems for which any $(\alpha_1, \ldots, \alpha_p)$-approximate Pareto set requires exponentially many solutions. This follows since even many biobjective optimization problems (e.g., the biobjective ShortestPath problem) admit instances with exponentially many different nondominated points (see, e.g., [7]). Thus, using the two given objective functions as the objectives f_i for two positions i with $\alpha_i = 1$ (and arbitrary objective functions for all other positions) yields an instance with p objectives where exponentially many solutions are required in any $(\alpha_1, \ldots, \alpha_p)$-approximate Pareto set.

In contrast to this, Papadimitriou and Yannakakis [14] show that if $\alpha_i > 1$ for all i, there always exists an $(\alpha_1, \ldots, \alpha_p)$-approximate Pareto set of polynomial cardinality. In this paper, we focus on the case where $\alpha_i = 1$ for exactly one i. Thus, we study approximate Pareto sets where, for any feasible solution x, there exists a solution in the approximate Pareto set that has value no worse than $f_i(x)$ in objective f_i and simultaneously achieves an approximation factor of $1+\varepsilon$ in all other objective functions for some $\varepsilon > 0$. For simplicity, we assume that the first objective f_1 is to be optimized exactly, i.e., that $\alpha_1 = 1$ and $\alpha_j = 1+\varepsilon$ for $j = 2, \ldots, p$.

Definition 4 (*One-exact ε-Pareto set*) For $\varepsilon > 0$, a $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-approximate Pareto set is called a *one-exact ε-Pareto set*.

A common way of dealing with multiobjective optimization problems are scalarizations, which turn the multiple objective functions into one objective function in some useful way. The resulting single objective optimization problem can be solved using known methods from single objective optimization and the obtained solution can then be used in the process of solving the multiobjective problem. One of the most common scalarization methods consists of putting some upper bound/budget on all objective functions but one, which is then minimized subject to the resulting budget constraints on the other objectives (see, e.g., [8]).

Definition 5 (*Budget-Constrained Problem* (Constrained)) The subproblem Constrained is the following: Given an instance (X, f) of a multiobjective minimization problem and bounds $B_i > 0$, $i = 2, \ldots, p$, for all objective functions except the first one, either answer that there does not exist a feasible solution $x' \in X$ with

$$f_i(x') \leq B_i, \qquad i = 2, \ldots, p,$$

or return a feasible solution that minimizes f_1 among all such solutions, i.e., return $x \in X$ with

$$f_1(x) = \text{OPT}_1(B_2, \ldots, B_p) := \min_{x' \in X} \left\{ f_1(x') : \; f_i(x') \leq B_i \text{ for } i = 2, \ldots, p \right\},$$
$$f_i(x) \leq B_i, \qquad i = 2, \ldots, p.$$

This scalarization via budget constraints is widely used both in practice and in the theoretical literature on multiobjective optimization even though the problem CONSTRAINED is hard to solve even for the biobjective versions of many relevant optimization problems such as SHORTESTPATH. However, there often exists a PTAS, i.e., a polynomial-time algorithm that finds an arbitrarily good approximation. The problem of finding a $(1+\delta)$-approximation for CONSTRAINED for some given value $\delta > 0$ is called the *restricted problem* [6]:

Definition 6 (*Restricted Problem* (RESTRICT$_\delta$)) For $\delta > 0$, the subproblem RESTRICT$_\delta$ is the following: Given an instance (X, f) of a multiobjective minimization problem and bounds $B_i > 0$, $i = 2, \ldots, p$, for all objective functions except the first one, either answer that there does not exist a feasible solution $x' \in X$ with

$$f_i(x') \leq B_i, \qquad i = 2, \ldots, p,$$

or return $x \in X$ with

$$\begin{aligned} &f_1(x) \leq (1+\delta) \cdot \text{OPT}_1(B_2, \ldots, B_p), \\ &f_i(x) \leq B_i, && i = 2, \ldots, p. \end{aligned}$$

An alternative way of circumventing the hardness of the budget-constrained problem is to consider solutions that violate the given bounds slightly, while requiring an objective value that is at least as good as the objective value of any solution that respects the bounds [6]:

Definition 7 (DUALRESTRICT$_\delta$) For $\delta > 0$, the subproblem DUALRESTRICT$_\delta$ is the following: Given an instance (X, f) of a multiobjective minimization problem and bounds $S_i > 0$, $i = 2, \ldots, p$, for all objectives except the first one, either answer that there does not exist a feasible solution $x' \in X$ with

$$f_i(x') \leq S_i, \qquad i = 2, \ldots, p,$$

or return $x \in X$ with

$$\begin{aligned} &f_1(x) \leq \text{OPT}_1(S_2, \ldots, S_p), \\ &f_i(x) \leq (1+\delta) \cdot S_i, && i = 2, \ldots, p. \end{aligned}$$

Note that, in an instance of DUALRESTRICT$_\delta$, the case might occur where there does not exist any feasible solution $x' \in X$ with $f_i(x') \leq S_i$ for $i = 2, \ldots, p$, but there exists a solution $x \in X$ with $f_i(x) \leq (1+\delta) \cdot S_i$ for $i = 2, \ldots, p$. In this case, NO is a correct answer to the DUALRESTRICT$_\delta$ instance, but, since $\text{OPT}_1(S_2, \ldots, S_p) = +\infty > f_1(x)$, also x is a correct answer. Thus, for DUALRESTRICT$_\delta$, there are situations where both of the distinguished cases apply, whereas the two considered cases are always disjoint for CONSTRAINED and RESTRICT$_\delta$. Also note that CONSTRAINED can be viewed as the limit case $\delta = 0$ for both RESTRICT$_\delta$ and DUALRESTRICT$_\delta$.

All three of the above subproblems can also be defined such that, instead of the first one, some other objective is to be optimized subject to budgets on the rest of the objectives. In the following, we sometimes use a superscript to indicate which objective is to be optimized. For example, RESTRICT$_\delta^i$ denotes the restricted problem with a bound on all objectives but the i-th one.

3 Existence and cardinality of one-exact ε-Pareto sets

Papadimitriou and Yannakakis [14] show that, for any instance of a multiobjective optimization problem, there exists an ε-Pareto set whose cardinality is polynomial in the encoding length of the instance and in $\frac{1}{\varepsilon}$. Similarly, we now show the existence of one-exact ε-Pareto sets of polynomial cardinality.

Theorem 1 *For any p-objective optimization problem (X, f) and any given $\varepsilon > 0$, there exists a one-exact ε-Pareto set of cardinality $\mathcal{O}\left(\left(\frac{M}{\varepsilon}\right)^{p-1}\right)$.*

Proof Figure 1 illustrates the proof.

Consider the hypercube $[2^{-M}, 2^M] \times \cdots \times [2^{-M}, 2^M]$, in which all the feasible points are contained, and cover this hypercube by hyperstripes of the form $[2^{-M}, 2^M] \times [(1+\varepsilon)^{i_2}, (1+\varepsilon)^{i_2+1}] \times \cdots \times [(1+\varepsilon)^{i_p}, (1+\varepsilon)^{i_p+1}]$, for all $i_2, \ldots, i_p \in \{-\lceil \frac{M}{\log(1+\varepsilon)} \rceil, \ldots, -1, 0, 1, \ldots, \lceil \frac{M}{\log(1+\varepsilon)} \rceil - 1\}$. Note that, for this covering, we use $\left(2 \cdot \lceil \frac{M}{\log(1+\varepsilon)} \rceil\right)^{p-1} = \mathcal{O}\left((\frac{M}{\varepsilon})^{p-1}\right)$ many hyperstripes.

For each hyperstripe H containing feasible points, we choose one feasible point $y = f(x) \in H$ with minimum f_1-value among all feasible points in H. Then all points in H are $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-dominated by y. Thus, keeping one solution $x \in X$ for each chosen point $y = f(x)$ (where points that are dominated by other chosen points can be discarded) yields a one-exact ε-Pareto set. Since at most one solution is chosen for each hyperstripe, the cardinality of the constructed set is in $\mathcal{O}\left((\frac{M}{\varepsilon})^{p-1}\right)$. □

It is easy to see that $(1, 1, 1+\varepsilon)$-approximate Pareto sets of polynomial cardinality do not exist in general since this would imply the existence of polynomial (exact) Pareto sets for biobjective optimization problems (see Remark 1). This means that, in this sense, an approximation factor of $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$ is the best one achievable with polynomially many solutions.

In general, even a smallest ε-Pareto set may require $\Omega\left((\frac{M}{\varepsilon})^{p-1}\right)$ many solutions [14], which equals the worst-case bound on the cardinality of a smallest one-exact ε-Pareto set obtained from Theorem 1. For some instances, however, the two can be very different in size. Any one-exact ε-Pareto set is, in particular, an ε-Pareto set, so, for any instance, a one-exact ε-Pareto set of minimum cardinality is at least as large as an ε-Pareto set of minimum cardinality. In the other direction, the following holds:

Theorem 2 *For any $\varepsilon > 0$ and any positive integer $n \in \mathbb{N}_+$, there exist instances of biobjective optimization problems such that $|P^*| > n \cdot |P^*_\varepsilon|$, where P^* denotes a smallest one-exact ε-Pareto set and P^*_ε denotes a smallest ε-Pareto set.*

Proof Given $\varepsilon > 0$ and $n \in \mathbb{N}_+$, we construct an instance of a biobjective minimization problem with $|P^*| = n+1$ and $|P^*_\varepsilon| = 1$.
Let $X := \{x_0, \ldots, x_n\}$ and, for $i = 0, \ldots, n$, let

$$f(x_i) = (f_1(x_i), f_2(x_i)) := \left(1 + \frac{n-i}{n} \cdot \varepsilon, (1+\varepsilon)^{2i}\right).$$

Then, we have

$$f_1(x_0) > f_1(x_1) > \ldots > f_1(x_n)$$

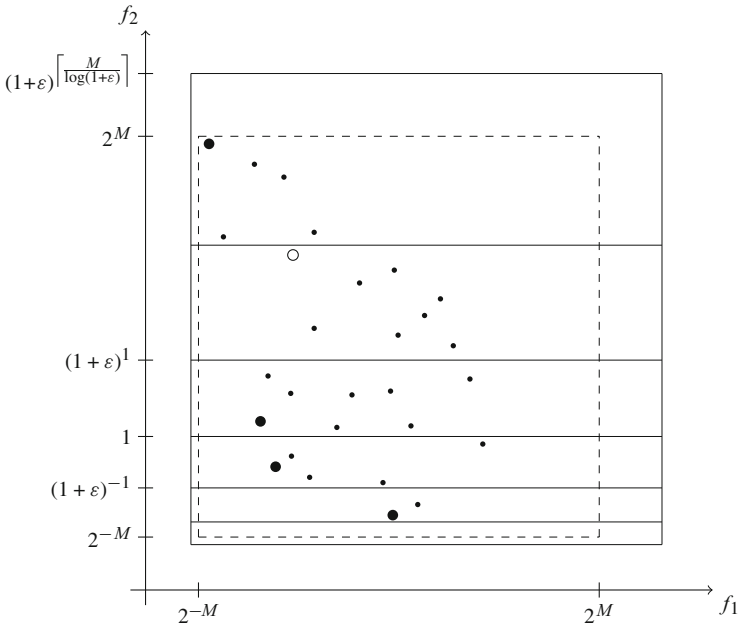

Fig. 1 Proof of existence of polynomial-cardinality one-exact ε-Pareto sets. Choose a feasible point minimizing f_1 in each hyperstripe that contains at least one feasible point. One can discard points that are dominated by other chosen points. In the picture, feasible points are marked by dots, chosen points are indicated by thick dots, and discarded points are drawn as circles

and

$$f_2(x_0) < \frac{1}{1+\varepsilon} \cdot f_2(x_1) < \ldots < \frac{1}{(1+\varepsilon)^n} \cdot f_2(x_n),$$

so no solution $(1, 1+\varepsilon)$-approximates any other solution and, thus, $P^* = X$. However, we also have

$$f_1(x_0) = 1 + \frac{n-0}{n} \cdot \varepsilon = 1 + \varepsilon = (1+\varepsilon) \cdot f_1(x_n),$$

so x_0 $(1+\varepsilon, 1+\varepsilon)$-approximates all other solutions and, thus, $\{x_0\}$ is an ε-Pareto set. This construction is depicted in Fig. 2. □

Note that, in the instance constructed in the proof of Theorem 2, we even have $|P^*| = \Omega\left(\frac{M}{\varepsilon}\right)$, i.e., a smallest one-exact ε-Pareto set, in fact, has the worst-case size, while a smallest one-exact ε-Pareto set P^*_ε consists of only one solution.

Moreover, the statement of Theorem 2 also holds for optimization problems with three or more objectives. For any $p \geq 2$, one can similarly construct instances of p-objective optimization problems where a smallest one-exact ε-Pareto set has the worst-case size of $\Omega\left((\frac{M}{\varepsilon})^{p-1}\right)$, while a smallest ε-Pareto set has cardinality one.

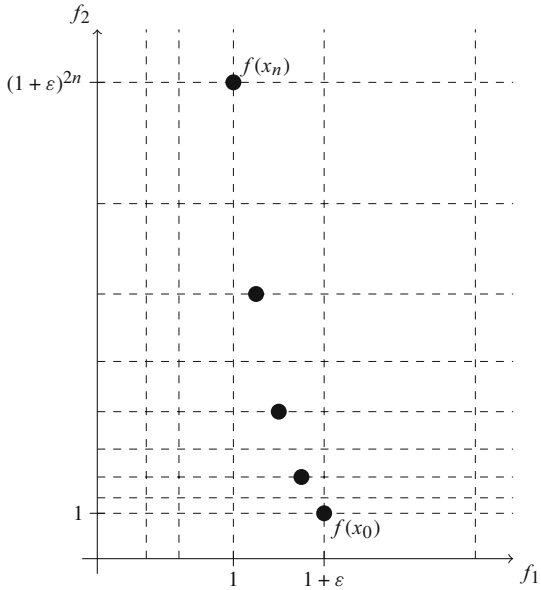

Fig. 2 Illustration of the instance in the proof of Theorem 2. The solution x_0 $(1+\varepsilon, 1+\varepsilon)$-approximates all other solutions, but no solution $(1, 1+\varepsilon)$-approximates any other solution

4 Polynomial-time computability of one-exact Pareto sets

The proof of Theorem 1 can easily be turned into a method for computing one-exact ε-Pareto sets that runs in fully polynomial time if and only if CONSTRAINED^1 is solvable in polynomial time. In the appendix, we present a method that, for biobjective optimization problems, even computes a smallest one-exact ε-Pareto set using a subroutine for CONSTRAINED. However, CONSTRAINED is NP-hard to solve for many relevant optimization problems.

Instead, we now provide a method that computes a one-exact ε-Pareto set in (fully) polynomial time if a (fully) polynomial method for DUALRESTRICT$^1_\delta$ is available. This is the case for a significantly larger class of (relevant) optimization problems including important problems such as multiobjective SHORTESTPATH and multiobjective SPANNINGTREE. The method is based on the following lemma, which is visualized in Fig. 3.

Lemma 1 *Let $x \in X$ be a solution to* DUALRESTRICT$^1_\delta(S_2, \ldots, S_p)$*, where $0 < \delta < \varepsilon$ for some $\varepsilon > 0$. Then any feasible point in the hyperstripe*

$$H = \left[2^{-M}, 2^M\right] \times \left[\frac{1+\delta}{1+\varepsilon} S_2, S_2\right] \times \cdots \times \left[\frac{1+\delta}{1+\varepsilon} S_p, S_p\right]$$

is $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-dominated by f(x).

Proof By the definition of DUALRESTRICT$_\delta$, we know that, in the first objective, we have $f_1(x) \leq \text{OPT}_1(S_2, \ldots, S_p)$, so there does not exist a feasible solution $x' \in X$ that satisfies $f_1(x') < f_1(x)$ and $f_i(x') \leq S_i$ for all $i = 2, \ldots, p$. We also know that $f_i(x) \leq (1+\delta) \cdot S_i$ for $i = 2, \ldots, p$.

Now, let $f(x') \in H$ be a feasible point in the hyperstripe. Then, since $f_i(x') \leq S_i$ for $i = 2, \ldots, p$, we must have $f_1(x') \geq f_1(x)$. Moreover, we have $f_i(x') \geq \frac{1+\delta}{1+\varepsilon} \cdot S_i$, which yields that

$$(1+\varepsilon) \cdot f_i(x') \geq (1+\varepsilon) \cdot \frac{1+\delta}{1+\varepsilon} \cdot S_i = (1+\delta) \cdot S_i \geq f_i(x),$$

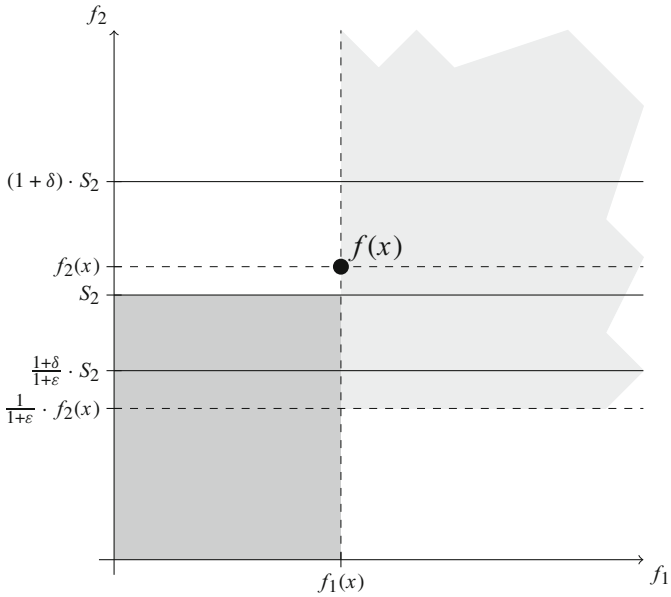

Fig. 3 Illustration of Lemma 1. The point $f(x)$ is the image of a solution $x \in X$ to $\text{DualRestrict}^1_\delta(S_2)$ with $0 < \delta < \varepsilon$. The dark gray region does not contain any feasible point. Every feasible point in the light gray region is $(1, 1+\varepsilon)$-dominated by $f(x)$

so $f(x')$ is $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-dominated by $f(x)$. □

Note that, if NO is a solution to $\text{DualRestrict}^1_\delta(S_2, \ldots, S_p)$, then the hyperstripe H considered in Lemma 1 does not contain any feasible point. Thus, we know a priori that solving $\text{DualRestrict}^1_\delta$ takes care of H in the sense that, if H contains feasible points, then $\text{DualRestrict}^1_\delta(S_2, \ldots, S_p)$ is guaranteed to yield a feasible solution $x \in X$ that $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-approximates every feasible solution $x' \in X$ with $f(x') \in H$.

If δ is chosen, e.g., such that $(1+\delta)^2 = 1+\varepsilon$, then the hypercube $[2^{-M}, 2^M]^p$, in which all feasible points are contained, can be covered by $\mathcal{O}\left((\frac{M}{\varepsilon})^{p-1}\right)$ many such hyperstripes, each of which, in turn, can be taken care of by one solution of $\text{DualRestrict}^1_\delta$.

This idea of covering the range of possible objective values by polynomially many solutions of $\text{DualRestrict}_\delta$ is used by Angel et al. [1,2] for the biobjective min-cost-makespan scheduling problem. We formalize the idea for general multiobjective optimization problems in Algorithm 1.

Algorithm 1: A $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-approximation for multiobjective optimization problems

input : an instance (X, f) of a p-objective minimization problem, $\varepsilon > 0$, an algorithm for $\textsc{DualRestrict}^1_\delta$, where $\delta = \sqrt{1+\varepsilon} - 1$
output: a one-exact ε-Pareto set for (X, f)

```
P ← ∅
δ ← √(1+ε) − 1
u ← ⌈ M / log(1+δ) ⌉
foreach (i_2, ..., i_p) such that i_ℓ ∈ {−u+1, ..., u}, ℓ = 1, ..., p do
    x ← DualRestrict^1_δ((1+δ)^{i_2}, ..., (1+δ)^{i_p})
    if x ≠ NO then
        P ← P ∪ {x}
return P
```

Theorem 3 *For a given instance (X, f) of a p-objective minimization problem and a given $\varepsilon > 0$, Algorithm 1 computes a one-exact ε-Pareto set. The algorithm solves $\mathcal{O}\left((\frac{M}{\varepsilon})^{p-1}\right)$ instances of* $\textsc{DualRestrict}^1_\delta$*, where $(1+\delta)^2 = 1+\varepsilon$. The returned set P has polynomial cardinality $|P| = \mathcal{O}\left((\frac{M}{\varepsilon})^{p-1}\right)$.*

Proof Algorithm 1 (implicitly) covers the hypercube $[2^{-M}, 2^M] \times \cdots \times [2^{-M}, 2^M]$ in the objective space by $(2u)^{p-1} = \left(2 \cdot \lceil \frac{M}{\log(1+\delta)} \rceil\right)^{p-1} = \mathcal{O}\left((\frac{M}{\varepsilon})^{p-1}\right)$ hyperstripes of the form

$$\begin{aligned} H &= \left[2^{-M}, 2^M\right] \times \left[(1+\delta)^{i_2-1}, (1+\delta)^{i_2}\right] \times \cdots \times \left[(1+\delta)^{i_p-1}, (1+\delta)^{i_p}\right] \\ &= \left[2^{-M}, 2^M\right] \times \left[\frac{1+\delta}{1+\varepsilon}(1+\delta)^{i_2}, (1+\delta)^{i_2}\right] \times \ldots \times \left[\frac{1+\delta}{1+\varepsilon}(1+\delta)^{i_p}, (1+\delta)^{i_p}\right]. \end{aligned}$$

Lemma 1 implies that, for each of these hyperstripes, solving the subproblem $\textsc{DualRestrict}^1_\delta((1+\delta)^{i_2}, \ldots, (1+\delta)^{i_p})$ either yields a feasible solution $x \in X$ such that all feasible points in the hyperstripe are $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-dominated by $f(x)$, or it yields NO, which guarantees that the hyperstripe does not contain any feasible point. Hence, the set of all feasible solutions produced by solving $\textsc{DualRestrict}^1_\delta$ for these hyperstripes is a one-exact ε-Pareto set of cardinality $\mathcal{O}\left((\frac{M}{\varepsilon})^{p-1}\right)$. □

We note that the one-exact ε-Pareto set returned by Algorithm 1 may contain solutions that are dominated by other solutions in the set. Such solutions can be removed without influencing the obtained approximation quality. However, filtering out dominated solutions might actually require more time than computing the set itself in situations where $\textsc{DualRestrict}_\delta$ can be solved very efficiently.

Papadimitriou and Yannakakis [14] show that there is an equivalence between solving the $\textsc{Gap}_\delta$ problem associated with a multiobjective optimization problem and finding an ε-Pareto set in the sense that one can compute an ε-Pareto set in (fully) polynomial time if and only if one can solve $\textsc{Gap}_\delta$ in (fully) polynomial time.

We now prove an analogous result for $\textsc{DualRestrict}_\delta$ and one-exact ε-Pareto sets. This demonstrates that $\textsc{DualRestrict}_\delta$ is, in fact, exactly the right auxiliary problem to consider for computing one-exact ε-Pareto sets.

Theorem 4 *A one-exact ε-Pareto set for an instance I of a multiobjective optimization problem can be found for any $\varepsilon > 0$ in time polynomial in the encoding length of I (and in $\frac{1}{\varepsilon}$) if and only if* DualRestrict$^1_\delta$ *can be solved for any $\delta > 0$ in time polynomial in the encoding length of I (and in $\frac{1}{\delta}$).*

Proof If DualRestrict$^1_\delta$ can be solved in (fully) polynomial time, a one-exact ε-Pareto set can be found in (fully) polynomial time using Algorithm 1.

Conversely, suppose that we can compute a one-exact ε-Pareto set in (fully) polynomial time. Then, given bounds $S_2, \ldots, S_p > 0$, we can solve DualRestrict$^1_\delta(S_2, \ldots, S_p)$ as follows:

We start by computing a one-exact δ-Pareto set P. Then, if there is no solution $x \in P$ with $f_i(x) \leq (1+\delta) \cdot S_i$ for $i = 2, \ldots, p$, we return NO. This is a correct answer since, if there was a solution x' with $f_i(x) \leq S_i$ for $i = 2, \ldots, p$, there would be no solution $x \in P$ that $(1, 1+\delta, \ldots, 1+\delta)$-approximates x' in contradiction to P being a one-exact δ-Pareto set. If there exist solutions $x \in P$ with $f_i(x) \leq (1+\delta) \cdot S_i$ for $i = 2, \ldots, p$, we return one of them with minimum value in f_1. Assume that, for the returned solution x, we have $f_1(x) > \text{OPT}_1(S_2, \ldots, S_p)$. Then this means that there is some $x' \in X$ with $f_i(x') \leq S_i$ for $i = 1, \ldots, p$ and

$$f_1(x') < f_1(x) = \min\{f_1(x'') : x'' \in P,\ f_i(x'') \leq (1+\delta) \cdot S_i, i = 2, \ldots, p\}.$$

Thus, x' is not $(1, 1+\delta, \ldots, 1+\delta)$-approximated by any solution in P, which again contradicts P being a one-exact δ-Pareto set. □

5 Computing small one-exact ε-Pareto sets

In this section, we consider the question if and how we can compute one-exact ε-Pareto sets that are not only of polynomial size, but also guarantee some bound on the cardinality compared to the cardinality of a smallest one-exact ε-Pareto set P^*.

The worst-case cardinality of a one-exact ε-Pareto set computed by Algorithm 1 is $\left(2 \cdot \lceil \frac{M}{\log(1+\delta)} \rceil\right)^{p-1}$ for $(1+\delta)^2 = 1+\varepsilon$, which is by a factor of 2^{p-1} larger than the upper bound of $\left(2 \cdot \lceil \frac{M}{\log(1+\varepsilon)} \rceil\right)^{p-1}$ for ε-Pareto sets constructed in the proof of Theorem 1. However, even when adding a filtering step that removes solutions dominated by other solutions in the computed set, Algorithm 1 does not provide an upper bound on the ratio $\frac{|P|}{|P^*|}$ for any fixed instance, where P is a one-exact ε-Pareto set computed by Algorithm 1 and P^* is a one-exact ε-Pareto set of minimum cardinality.

With such an additional filtering step, it is possible to show that, for biobjective optimization problems, we have an upper bound of 4 on the ratio $\frac{|P|}{|P^*|}$. When using $\delta = \sqrt[3]{1+\varepsilon} - 1$ instead of $\delta = \sqrt{1+\varepsilon} - 1$ in Algorithm 1 and replacing $1+\delta$ by $(1+\delta)^2$ in lines 3 and 5, we can improve this ratio to 3. We can even achieve a ratio of 2 when setting $\delta = \sqrt[4]{1+\varepsilon} - 1$ and using a more sophisticated elimination technique than simply filtering out dominated solutions.

Here, however, we derive a different algorithm for biobjective optimization problems that does not operate on a predefined grid but instead uses adaptive steps in order to decrease the number of solved instances of DualRestrict$_\delta$ while still ensuring a cardinality guarantee of $\frac{|P|}{|P^*|} \leq 2$ even without an additional (potentially time-consuming) filtering step.

We first give some results that substantiate the hardness of computing one-exact ε-Pareto sets that are smaller than twice the minimum size. Then, we formulate our algorithm and prove its correctness. We additionally consider the cases that an efficient routine for solving RESTRICT_δ or CONSTRAINED is given. Finally, we prove a result that indicates the hardness of achieving similar results for more than two objectives.

5.1 Lower bounds for biobjective optimization problems

The following result shows that, for biobjective optimization problems, any generic algorithm based on solving $\text{DUALRESTRICT}^1_\delta$ that computes a one-exact ε-Pareto set P of cardinality $|P| < 2 \cdot |P^*|$ needs to solve an instance of $\text{DUALRESTRICT}^1_\delta$ for some $\delta > 0$ for which $\frac{1}{\delta}$ is exponential in the encoding length of the input. Since the running time of a method for solving $\text{DUALRESTRICT}^1_\delta$ is typically at least linear in $\frac{1}{\delta}$, this implies that it is unlikely for such an algorithm to run in polynomial time. Note that, for optimization problems where the running time of a routine for $\text{DUALRESTRICT}^1_\delta$ is at most logarithmic in $\frac{1}{\delta}$, we can solve CONSTRAINED^1 efficiently by setting $\delta < 2^{-2M}$ in $\text{DUALRESTRICT}^1_\delta$. Thus, in this case, we can even compute a smallest one-exact ε-Pareto set in polynomial time (see Corollary 1).

Theorem 5 *For any $\varepsilon > 0$, there does not exist an algorithm that computes a one-exact ε-Pareto set P such that $|P| < 2 \cdot |P^*|$ for every biobjective optimization problem and generates feasible solutions only via solving* $\text{DUALRESTRICT}^1_\delta$ *for values of δ such that $\frac{1}{\delta}$ is polynomial in the encoding length of the input.*

Proof Given $\varepsilon > 0$, consider the two instances $I_1 = (\{x_1, x_2\}, f)$ and $I_2 = (\{x_1, x_2, x_3\}, f)$ in which $f(x_1) = (f_1(x_2) - 1, (1+\varepsilon) \cdot f_2(x_2))$ and $f(x_3) = (f_1(x_2), f_2(x_2) - 1)$. Then $\{x_1\}$ is a one-exact ε-Pareto set for I_1. On the other hand, any one-exact ε-Pareto set for I_2 needs at least two solutions since neither x_2 nor x_3 $(1, 1+\varepsilon)$-approximates x_1, and x_1 does not $(1, 1+\varepsilon)$-approximate x_3. An algorithm that computes a one-exact ε-Pareto set P with $|P| < 2 \cdot |P^*|$ would, therefore, have to be able to distinguish between I_1 and I_2, i.e., detect the existence of x_3.

Note that, for $S_2 < f_2(x_3)$, NO is a solution to $\text{DUALRESTRICT}^1_\delta(S_2)$ in both instances I_1 and I_2 for any δ. If $f_2(x_1) > S_2 \geq f_2(x_3)$ and $\delta \geq \frac{1}{f_2(x_3)}$, we have

$$f_2(x_2) = f_2(x_3) + 1 = \left(1 + \frac{1}{f_2(x_3)}\right) \cdot f_2(x_3) \leq (1+\delta) \cdot f_2(x_3) \leq (1+\delta) \cdot S_2,$$

so x_2 is a solution to $\text{DUALRESTRICT}^1_\delta(S_2)$ in both instances. For $S_2 \geq f_2(x_1)$, x_1 is a solution to $\text{DUALRESTRICT}^1_\delta(S_2)$ in both instances for any δ. Therefore, in order to tell the difference between I_1 and I_2, an algorithm using only $\text{DUALRESTRICT}^1_\delta$ to generate feasible solutions would have to solve $\text{DUALRESTRICT}^1_\delta(S_2)$ for S_2 and δ with $f_2(x_1) > S_2 \geq f_2(x_3)$ and $\delta < \frac{1}{f_2(x_3)}$, i.e., $\frac{1}{\delta} > f_2(x_3) = f_2(x_2) - 1$. Since the value $f_2(x_2)$ might be exponentially large in the encoding length of the input, this proves the claim. □

While Theorem 5 shows that generic algorithms based on $\text{DUALRESTRICT}_\delta$ cannot obtain a factor smaller than 2 with respect to the cardinality of a one-exact ε-Pareto set without using exponentially large values of $\frac{1}{\delta}$ (even if $\mathsf{P} = \mathsf{NP}$), we now show that, for certain biobjective optimization problems, *no* algorithm (whether based only on $\text{DUALRESTRICT}_\delta$ or not) can obtain a factor smaller than 2 under the assumption that $\mathsf{P} \neq \mathsf{NP}$. To this end, we consider the following biobjective scheduling problem: We are given a set J of $|J| = n$ independent

jobs, which are to be scheduled on m parallel machines. Performing job j on machine i takes processing time $p_{ij} \geq 0$ and causes cost $c_{ij} \geq 0$. The goal is to minimize the makespan (i.e., the maximum completion time of a job) and the total cost (i.e., the sum of the costs resulting from assigning jobs to machines). We call this biobjective optimization problem, where the cost-objective is the first objective f_1 and the makespan-objective is the second objective f_2, the *min-cost-makespan scheduling problem*.

The min-cost-makespan scheduling problem is known to have a fully polynomial-time algorithm for DualRestrict$^1_\delta$ and a fully polynomial-time one-exact approximation algorithm due to Angel et al. [1,2]. For this problem, however, we can show the following additional hardness result regarding the computation of small one-exact ε-Pareto sets.

Theorem 6 *For the min-cost-makespan scheduling problem, if* $0 < \varepsilon < \frac{1}{2}$, *it is NP-hard to compute a one-exact* ε*-Pareto set* P *of cardinality* $|P| < 2 \cdot |P^*|$.

Proof We use a reduction from Partition. Given an instance $a_1, \ldots, a_n \in \mathbb{N}$ of Partition, where, without loss of generality, $A := \sum_{i=1}^n a_i \geq \frac{4}{1-2\varepsilon}$, define an instance of the min-cost-makespan scheduling problem as follows: We have $m = 2$ machines and $|J| = n + 2$ jobs. For $j = 1, \ldots, n$, we have a job j with processing times $p_{1j} = p_{2j} = a_j$ and costs $c_{1j} = a_j$ and $c_{2j} = 0$. We have two additional jobs $n + 1$ and $n + 2$, with $p_{1(n+1)} = p_{1(n+2)} = p_{2(n+1)} = p_{2(n+2)} = K$, $c_{1(n+1)} = c_{2(n+2)} = 1$, and $c_{2(n+1)} = c_{1(n+2)} = 2$, where $K > 0$ is chosen such that

$$\frac{1}{1+\varepsilon} \cdot (K + A) > K + \frac{A}{2}, \tag{1}$$

$$\frac{1}{1+\varepsilon} \cdot (K + A) \leq K + \frac{A}{2} + 1, \tag{2}$$

$$\frac{1}{1+\varepsilon} \cdot (K + A) \leq 2 \cdot K. \tag{3}$$

Note that it is possible to choose K like this by our assumptions on ε and A. For instance, one can check that $K := \lceil \frac{1-\varepsilon}{2\varepsilon} \cdot A - 1 - \frac{1}{\varepsilon} \rceil$ fulfills (1)–(3) and $K > 0$.

The schedule $\bar{s}$ where machine 1 performs only $\{n + 1\}$ and machine 2 performs the jobs $\{1, \ldots, n, n + 2\}$ has a cost of $f_1(\bar{s}) = 2$. This is the unique minimum in f_1 over all schedules, so the schedule $\bar{s}$ is not $(1, 1 + \varepsilon)$-approximated by any other schedule. Thus, it must be part of every one-exact ε-Pareto set. Moreover, $\bar{s}$ has a makespan of $f_2(\bar{s}) = K + A$, so, by inequality (3), it $(1, 1 + \varepsilon)$-approximates every schedule where jobs $n + 1$ and $n + 2$ are performed on the same machine.

If the instance of Partition is a NO-instance, any schedule where jobs $n+1$ and $n+2$ are performed on different machines has a makespan of at least $K + \frac{A}{2} + 1$, so, by inequality (2), the one-element set $\{\bar{s}\}$ is a one-exact ε-Pareto set.

If the instance of Partition is a YES-instance, i.e., if there exists a partition (I_1, I_2) such that $\sum_{i \in I_1} a_i = \sum_{i \in I_2} a_i = \frac{A}{2}$, then the schedule where machine 1 performs jobs $\{n+1\} \cup I_1$ and machine 2 performs jobs $\{n + 2\} \cup I_2$ has a makespan of $K + \frac{A}{2}$. By inequality (1), this schedule is not $(1, 1 + \varepsilon)$-approximated by $\bar{s}$, so any one-exact ε-Pareto set must contain at least two solutions. Therefore, it is NP-hard to distinguish between the two cases $|P^*| = 1$ and $|P^*| \geq 2$. □

Algorithm 2: A $(1, 1+\varepsilon)$-approximation for biobjective optimization problems

input : an instance (X, f) of a biobjective minimization problem, $\varepsilon > 0$, an algorithm for $\textsc{DualRestrict}^1_\delta$, where $\delta = \sqrt[4]{1+\varepsilon} - 1$
output: a one-exact ε-Pareto set for (X, f)

```
P ← ∅
δ ← ⁴√(1 + ε) − 1
S2 ← 2^M
repeat
    x ← DualRestrict^1_δ(S2)
    if x = NO then break(repeat)
    S2 ← 1/(1+δ)^2 · f2(x)
    x^next ← DualRestrict^1_δ(S2)
    if x^next = NO then P ← P ∪ {x} and break(repeat)
    while f1(x^next) = f1(x) do
        x ← x^next
        S2 ← 1/(1+δ)^2 · f2(x)
        x^next ← DualRestrict^1_δ(S2)
        if x^next = NO then P ← P ∪ {x} and break(while,repeat)
    P ← P ∪ {x}
    S2 ← 1/(1+ε) · f2(x)
return P
```

5.2 Algorithm for biobjective optimization problems

We now provide an algorithm that computes a one-exact ε-Pareto set P that is not larger than twice the cardinality of a smallest one-exact ε-Pareto set P^*. It is formally stated in Algorithm 2 and an illustration of its behavior is given in Fig. 4. The algorithm explores the objective space top-left to bottom-right, i.e., in decreasing values of f_2 and increasing values of f_1. It solves instances of $\textsc{DualRestrict}^1_\delta$ for $\delta > 0$ such that $(1+\delta)^4 = 1+\varepsilon$ and starts by solving $\textsc{DualRestrict}^1_\delta(2^M)$. This yields a solution that has minimum f_1-value among all feasible solutions. However, this solution is not automatically added to the output set. Instead, the algorithm checks whether there exists a solution with the same f_1-value and an f_2-value that is better by at least a factor of $(1+\delta)^2$ by solving $\textsc{DualRestrict}^1_\delta(S_2)$, where S_2 is chosen as the f_2-value of the previous solution divided by $(1+\delta)^2$ (we say that the algorithm does a small (multiplicative) step of $(1+\delta)^2 = \sqrt{1+\varepsilon}$). If the newly-found solution has the same f_1-value as the previous solution, the algorithm discards the previous one and repeats the small step with the new solution. If the new solution has a larger f_1-value than the previous one, there cannot exist any solution that has the same (or a better) f_1-value as the previous solution and is better in f_2 by at least a factor of $(1+\delta)^2$. Thus, the previous solution is added to the output set.

The added solution $(1, 1+\varepsilon)$-approximates all solutions that have the same or a worse f_1-value and an f_2-value that is better by at most a factor of $1+\varepsilon$. Therefore, instead of resuming with the new solution (which resulted from a small step) the algorithm resumes by solving $\textsc{DualRestrict}^1_\delta(S_2)$, where S_2 is chosen as the f_2-value of the added solution divided by $1+\varepsilon$ (we say that the algorithm does a large (multiplicative) step of $1+\varepsilon$). This ensures that the next solution has an f_2-value that is better than the f_2-value of the added

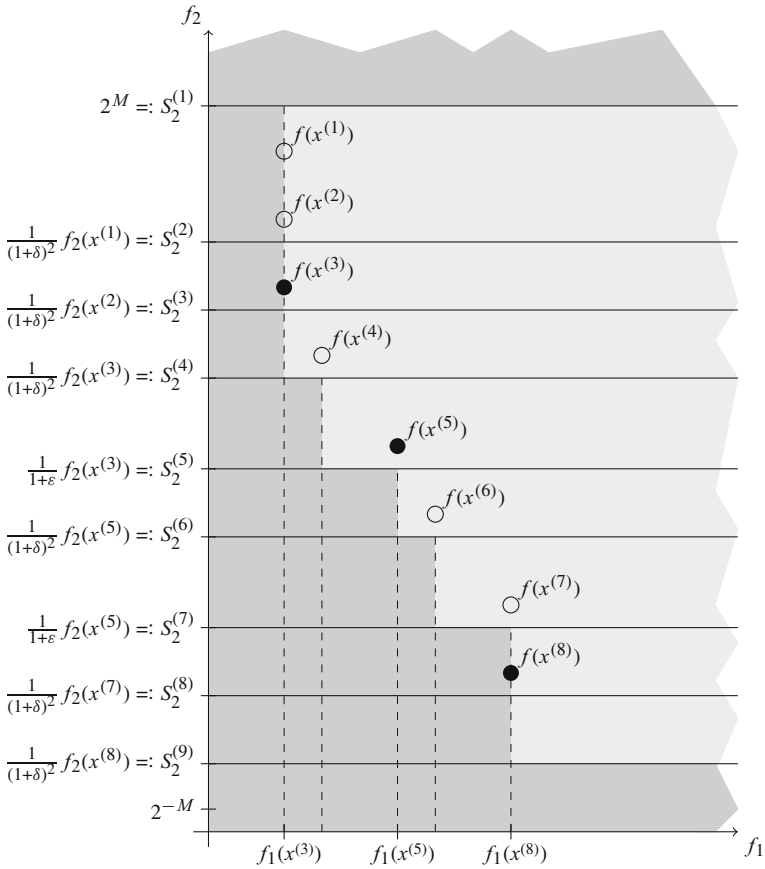

Fig. 4 An illustration of Algorithm 2 in the objective space (on a logarithmic scale). Each $x^{(i)}$ is a solution to $\textsc{DualRestrict}^1_\delta(S_2^{(i)})$ for $i = 1, \ldots, 8$, where $(1+\delta)^4 = (1+\varepsilon)$. The dark gray region does not contain any feasible point. Any feasible point in the light gray region is $(1, 1+\varepsilon)$-dominated by $f(x^{(i)})$ for some $i \in \{1, \ldots, 8\}$. The solution $x^{(4)}$ is discarded since any solution that is $(1, 1+\varepsilon)$-approximated by $x^{(4)}$ is also $(1, 1+\varepsilon)$-approximated by $x^{(3)}$ or $x^{(5)}$. The solution $x^{(6)}$ is discarded since any solution that is $(1, 1+\varepsilon)$-approximated by $x^{(6)}$ is also $(1, 1+\varepsilon)$-approximated by $x^{(5)}$ or $x^{(7)}$. The solutions $x^{(1)}$, $x^{(2)}$, and $x^{(7)}$ are discarded since they are dominated by $x^{(2)}$, $x^{(3)}$, and $x^{(8)}$, respectively. $\textsc{DualRestrict}_\delta(S_2^{(9)})$ returns NO, so the algorithm returns $\{x^{(3)}, x^{(5)}, x^{(8)}\}$

solution by at least a factor of $(1+\delta)^3$. This adaptive procedure of small and large steps is iterated until $\textsc{DualRestrict}^1_\delta(S_2)$ yields NO.

The following lemma collects several invariants that hold during the execution of Algorithm 2.

Lemma 2 *When performing line 10 in any iteration of the while/repeat loops in Algorithm 2, the following properties hold:*

(a) $x \neq \text{NO}$ *and* $x^{next} \neq \text{NO}$,
(b) $S_2 = \frac{1}{(1+\delta)^2} \cdot f_2(x)$,
(c) $f_1(x^{next}) \leq \text{OPT}_1(S_2)$,

(d) $f_2(x^{next}) \leq \frac{1}{1+\delta} \cdot f_2(x)$,
(e) $f_1(x^{next}) \geq f_1(x)$,
(f) *The solutions in* $P \cup \{x\}$ *do* $(1, 1+\varepsilon)$*-approximate all solutions* $x' \in X$ *with* $f_2(x') \geq \frac{1}{1+\varepsilon} \cdot f_2(x)$,
(g) $f_2(x) \leq \frac{1}{(1+\delta)^3} \cdot \min\{f_2(\bar{x}) : \bar{x} \in P\}$.

Proof. (a) As soon as solving an instance of $\textsc{DualRestrict}^1_\delta$ yields NO during the execution of Algorithm 2, the repeat loop breaks immediately, so line 10 is not reached anymore.
(b) Before line 10, either line 7 or line 12 is executed.
(c) Before line 10, either line 8 or line 13 is executed. Therefore we have $x^{\text{next}} = \textsc{DualRestrict}^1_\delta(S_2)$, which implies $f_1(x^{\text{next}}) \leq \textsc{opt}_1(S_2)$.
(d) Again, we have $x^{\text{next}} = \textsc{DualRestrict}^1_\delta(S_2)$. This implies that $f_2(x^{\text{next}}) \leq (1+\delta) \cdot S_2$. Using (b), we obtain that $f_2(x^{\text{next}}) \leq (1+\delta) \cdot \frac{1}{(1+\delta)^2} \cdot f_2(x) = \frac{1}{1+\delta} \cdot f_2(x)$.
(e) Since the only way feasible solutions are generated in Algorithm 2 is by solving $\textsc{DualRestrict}^1_\delta$, x is a solution to $\textsc{DualRestrict}^1_\delta(S)$ for some parameter $S \in \mathbb{Q}$, where $f_1(x) \leq \textsc{opt}_1(S)$ and $f_2(\tilde{x}) \leq (1+\delta) \cdot S$. Due to this and (c), we have

$$f_2(x^{\text{next}}) \leq \frac{1}{1+\delta} \cdot f_2(x) \leq \frac{1}{1+\delta} \cdot (1+\delta) \cdot S = S,$$

so we can conclude that $f_1(x) \leq \textsc{opt}_1(S) \leq f_1(x^{\text{next}})$.
(f) Consider an iteration of the inner while loop and suppose that (a)–(f) hold at the beginning of this iteration. We show that (f) also holds at the end of this iteration (provided that the algorithm does not terminate during the iteration). At the beginning of the iteration, we have $f_1(x^{\text{next}}) = f_1(x)$, so (d) implies that x^{next} strictly dominates x. Therefore, also $P \cup \{x^{\text{next}}\}$ approximates all solutions $x' \in X$ with $f_2(x') \geq \frac{1}{1+\varepsilon} \cdot f_2(x)$. Moreover, using (c) and (e), we know that

$$f_1\left(x^{\text{next}}\right) \leq \textsc{opt}_1(S_2) = \textsc{opt}_1\left(\frac{1}{(1+\delta)^2} \cdot f_2(x)\right) \leq \textsc{opt}_1\left(\frac{1}{1+\varepsilon} \cdot f_2(x)\right),$$

so x^{next} also approximates all solutions $x' \in X$ with $\frac{1}{1+\varepsilon} \cdot f_2(x) > f_2(x') \geq \frac{1}{1+\varepsilon} \cdot f_2(x^{\text{next}})$. During the iteration of the while loop, line 11 is performed, which implies that (f) holds at the end of the iteration.
Now consider an iteration of the outer repeat loop and assume that (a)–(f) hold in line 15 of this iteration. We show that (f) holds in line 10 of the next iteration (given that line 10 is reached). After performing line 16, we know that P approximates every solution $x' \in X$ with $f_2(x') \geq \frac{1}{1+\varepsilon} \cdot f_2(x) = S_2$. This implies that (f) holds after line 5 of the next iteration and, thus, also in line 10 of the next iteration.
(f) If (d) holds at the beginning of a fixed iteration of the while loop, then the value of $f_2(x)$ decreases during the iteration. Therefore, (g) holding at the beginning of the iteration together with the fact that P remains unchanged during the while loop imply that (g) also holds at the end of the iteration.
If (g) holds in line 15 of an iteration of the repeat loop, then, after line 16, we have that $\min\{f_2(\bar{x}) : \bar{x} \in P\} = f_2(x)$, so $S_2 = \frac{1}{1+\varepsilon} \cdot \min\{f_2(\bar{x}) : \bar{x} \in P\}$ holds after line 16. After line 5 in the next iteration, we thus have

$$f_2(x) \leq (1+\delta) \cdot S_2 = \frac{1}{(1+\delta)^3} \cdot \min\{f_2(\bar{x}) : \bar{x} \in P\}.$$

□ □

The following two results establish a bound on the number of instances of $\text{DualRestrict}^1_\delta$ solved in the algorithm.

Lemma 3 *Every time the value of S_2 is changed from some value S_2^{old} to a new value S_2^{new} during the execution of Algorithm 2, we have*

$$S_2^{\text{new}} \leq \frac{1}{1+\delta} \cdot S_2^{\text{old}}.$$

Proof. We distinguish three cases: If the value of S_2 is changed from S_2^{old} to S_2^{new} in line 7, we have

$$S_2^{\text{new}} = \frac{1}{(1+\delta)^2} \cdot f_2(x) \leq \frac{1}{(1+\delta)^2} \cdot (1+\delta) \cdot S_2^{\text{old}} = \frac{1}{1+\delta} \cdot S_2^{\text{old}},$$

where the inequality is due to line 5.

If the value of S_2 is changed from S_2^{old} to S_2^{new} in line 12, Lemma 2 (b) and (d) imply that

$$S_2^{\text{new}} = \frac{1}{(1+\delta)^2} \cdot f_2(x) = \frac{1}{(1+\delta)^2} \cdot f_2(x^{\text{next}}) \leq \frac{1}{1+\delta} \cdot S_2^{\text{old}}.$$

If the value of S_2 is changed from S_2^{old} to S_2^{new} in line 16, we have $S_2^{\text{new}} = \frac{1}{1+\varepsilon} \cdot f_2(x)$. Since Lemma 2 (b) yields that $S_2^{\text{old}} = \frac{1}{(1+\delta)^2} \cdot f_2(x)$, this implies that

$$S_2^{\text{new}} = \frac{1}{(1+\delta)^2} \cdot S_2^{\text{old}} < \frac{1}{1+\delta} \cdot S_2^{\text{old}}.$$

□ □

Proposition 1 *Algorithm 2 terminates after solving $\mathcal{O}\left(\frac{M}{\varepsilon}\right)$ many instances of* $\text{DualRestrict}^1_\delta$.

Proof As soon as solving an instance of $\text{DualRestrict}^1_\delta$ yields NO, Algorithm 2 terminates. Note that $\text{DualRestrict}^1_\delta(S_2)$ is solved once whenever S_2 is assigned a new value. We show that the number of times the value of S_2 is changed is bounded by $\left\lfloor \frac{2M}{\log(1+\delta)} \right\rfloor + 1 = \mathcal{O}\left(\frac{M}{\varepsilon}\right)$. The initial value assigned to S_2 is 2^M. $\text{DualRestrict}^1_\delta(S_2)$ yields NO if $S_2 < 2^{-M}$, so if $\text{DualRestrict}^1_\delta(S_2)$ is solved for some $S_2 < 2^{-M}$, the algorithm is guaranteed to terminate. But this is the case after at most $\left\lfloor \frac{2M}{\log(1+\delta)} \right\rfloor + 1$ many changes of the value assigned to S_2 due to Lemma 3. □

The correctness of the algorithm is established by the following proposition.

Proposition 2 *Algorithm 2 returns a one-exact ε-Pareto set.*

Proof Algorithm 2 terminates as soon as solving $\text{DualRestrict}^1_\delta(S_2)$ yields NO for some S_2. We do a case distinction on where this happens in Algorithm 2 and show that the solutions in the set P returned by the algorithm $(1, 1+\varepsilon)$-approximate all solutions $x' \in X$ with $f_2(x') \geq S_2$ (and, thus, all solutions $x' \in X$). If $\text{DualRestrict}^1_\delta(S_2)$ yields NO in line 5 of the first iteration of the repeat loop, this implies that there does not exist any feasible solution at all. If $\text{DualRestrict}^1_\delta(S_2)$ yields NO in line 5 of some other iteration of the repeat loop, Lemma 2 (f) holds in line 10 of the previous iteration, in which, in particular, lines 15 and 16 are executed. Therefore, the solutions in the set P returned by Algorithm 2 approximate all solutions $x' \in X$ with $f_2(x') \geq \frac{1}{1+\varepsilon} \cdot f_2(x) = S_2$. If $\text{DualRestrict}^1_\delta(S_2)$ yields NO in line 8 or line 13 of some iteration (of the repeat loop or while loop, respectively), we can

show that Lemma 2 (f) holds at this moment by using the same argumentation as in the proof of Lemma 2 (f). Therefore and since x is added to P in these two cases, the solutions in the returned set P approximate all solutions $x' \in X$ with $f_2(x') \geq \frac{1}{1+\varepsilon} \cdot f_2(x) > \frac{1}{(1+\delta)^2} \cdot f_2(x) = S_2$. □

It remains to bound the cardinality of the returned one-exact ε-Pareto set P. To this end, we first show in Lemma 4 that any two solutions in P always differ by at least a factor of $(1+\delta)^3$ with respect to their f_2-values.

Lemma 4 *Let P be the set returned by Algorithm 2. For all $x_1, x_2 \in P$ with $x_1 \neq x_2$ and $f_2(x_1) \geq f_2(x_2)$, we have*

$$f_2(x_1) \geq (1+\delta)^3 \cdot f_2(x_2),$$

where $(1+\delta)^4 = 1+\varepsilon$.

Proof Note that Lemma 2 (g) holds each time a solution is added to P in line 15 of Algorithm 2. If some solution $x \in X$ is added to P in line 9 or line 14, $\textsc{DualRestrict}^1_\delta(S_2)$ must have yielded NO in line 8 or line 13, respectively. We can show that Lemma 2 (g) holds at this moment by using the same argumentation as in the proof of Lemma 2 (g). □

Lemma 5 establishes that solutions in P are almost efficient in the sense that no other solution with the same or a better f_1-value is better by a factor of $(1+\delta)^2$ or more in f_2.

Lemma 5 *Let P be the set returned by Algorithm 2. For any $x \in P$, there does not exist any feasible solution $x' \in X$ with $f_1(x') \leq f_1(x)$ and $f_2(x') \leq \frac{1}{(1+\delta)^2} \cdot f_2(x)$.*

Proof Given $x \in P$, consider the case that x is added to P in line 15 of Algorithm 2. At this point in the algorithm, we know that $f_1(x^{\text{next}}) \neq f_1(x)$. Thus, Lemma 2 (b), (c), and (e) imply that $f_1(x) < f_1(x^{\text{next}}) \leq \textsc{opt}_1(S_2) = \textsc{opt}_1(\frac{1}{(1+\delta^2)} \cdot f_2(x))$, so any solution $x' \in X$ with $f_2(x') \leq \frac{1}{(1+\delta^2)} \cdot f_2(x)$ must have $f_1(x') > f_1(x)$.

Now consider the case that x is added to P in line 9 or line 14 of Algorithm 2. In this case, we have $\textsc{DualRestrict}^1_\delta(S_2) = \text{NO}$ for $S_2 = \frac{1}{(1+\delta^2)} \cdot f_2(x)$, so there does not exist any solution $x' \in X$ with $f_2(x') \leq \frac{1}{(1+\delta^2)} \cdot f_2(x)$. □

Proposition 3 *Let P be the set returned by Algorithm 2 and let P^* be a smallest one-exact ε-Pareto set. Then*

$$|P| \leq 2 \cdot |P^*|.$$

Proof First, we show that no solution $x' \in X$ can $(1, 1+\varepsilon)$-approximate more than two solutions in the returned set P. Let $x_1, x_2, x_3 \in P$ be three pairwise different solutions in P such that $f_2(x_1) \geq f_2(x_2) \geq f_2(x_3)$. Let $x' \in X$ be an arbitrary feasible solution that $(1, 1+\varepsilon)$-approximates x_1 and x_2. We show that x' does not $(1, 1+\varepsilon)$-approximate x_3. Note that Lemma 4 implies that

$$f_2(x_3) \leq \frac{1}{(1+\delta)^3} \cdot f_2(x_2) \leq \frac{1}{(1+\delta)^6} \cdot f_2(x_1).$$

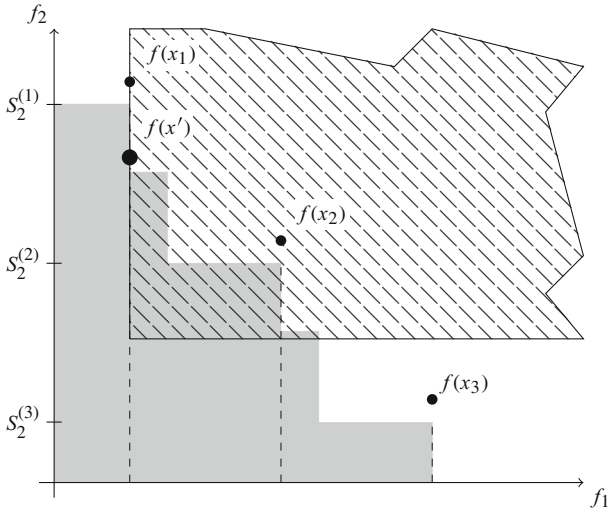

Fig. 5 Any feasible solution $x' \in X$ can $(1, 1+\varepsilon)$-approximate at most two solutions returned by Algorithm 2. The solution x_i is a solution to DualRestrict$^1_\delta(S_2^{(i)})$ for $i = 1, 2, 3$, where $(1+\delta)^4 = (1+\varepsilon)$. The gray region does not contain any feasible point due to Lemma 5 and the definition of DualRestrict$_\delta$, so $f(x')$ has to lie outside of this region. The hatched region is the region that is $(1, 1+\varepsilon)$-dominated by $f(x')$

Since x_1 is $(1, 1+\varepsilon)$-approximated by x', we have $f_1(x') \leq f_1(x_1)$, so Lemma 5 implies that

$$f_2(x') > \frac{1}{(1+\delta)^2} \cdot f_2(x_1).$$

Combining the two inequalities above yields

$$f_2(x') > (1+\delta)^4 \cdot f_2(x_3) = (1+\varepsilon) \cdot f_2(x_3),$$

i.e., x_3 is not $(1, 1+\varepsilon)$-approximated by x'. An illustration of this is given in Fig. 5. Note that the above arguments also imply that no solution $x_0 \in P$ with $x_0 \neq x_1$ and $f_2(x_0) \geq f_2(x_1)$ can be approximated by x'. If this was the case, then x' would not approximate x_2.

Now let P^* be an arbitrary minimum-cardinality one-exact ε-Pareto set. Then, for any $x \in P$, there exists some $x' \in P^*$ that $(1, 1+\varepsilon)$-approximates x. Thus, since any $x' \in P^*$ can approximate at most two solutions in P, there have to be at least $\left\lceil \frac{|P|}{2} \right\rceil$ many elements in P^*, so $|P| \leq 2 \cdot \left\lceil \frac{|P|}{2} \right\rceil \leq 2 \cdot |P^*|$. □

Propositions 1, 2, and 3 directly yield the following theorem:

Theorem 7 *Algorithm 2 computes a one-exact ε-Pareto set P of cardinality $|P| \leq 2 \cdot |P^*|$ solving $\mathcal{O}(\frac{M}{\varepsilon})$ many instances of* DualRestrict$^1_\delta$, *where* $(1+\delta)^4 = 1+\varepsilon$.

5.3 Available efficient routine for Restrict$_\delta$

Diakonikolas and Yannakakis [6] show that, for biobjective optimization problems, the subproblems DualRestrict$^1_\delta$ and Restrict$^2_\delta$ are polynomially equivalent: An answer for an

instance of $\text{DualRestrict}^1_\delta$ can be found by solving $\mathcal{O}(M)$ instances of Restrict^2_δ in a binary search and vice versa.

They also give an algorithm that computes an ε-Pareto set P_ε whose cardinality is not larger than twice the cardinality of a smallest ε-Pareto set P^*_ε. This algorithm is based on routines for both of these subproblems. In order to compute P_ε, the algorithm solves $\mathcal{O}(\frac{M}{\varepsilon})$ instances of $\text{DualRestrict}^1_\delta$ as well as $\mathcal{O}(\frac{M}{\varepsilon})$ instances of Restrict^2_δ, where $(1+\delta)^3 = 1+\varepsilon$. This a-priori bound can be refined a posteriori to the output-sensitive[3] bound of $\mathcal{O}(|P_\varepsilon|) = \mathcal{O}(|P^*_\varepsilon|)$ many instances of each subproblem that are solved. If only one of the two routines is directly available (such as, e.g., for the min-cost makespan scheduling problem), the polynomial reduction between the two subproblems can be used to solve the other problem in the algorithm by Diakonikolas and Yannakakis, resulting in $\mathcal{O}(\frac{M^2}{\varepsilon})$ (or a posteriori $\mathcal{O}(M \cdot |P^*_\varepsilon|)$) solved instances of the subproblem for which a routine is available.

We now show how this algorithm can be slightly modified such that it even computes a one-exact ε-Pareto set in the same a-priori asymptotic running time. The cardinality of the one-exact ε-Pareto set Q computed by this modified algorithm satisfies $|Q| \leq 2 \cdot |P^*|$, where P^* is a smallest one-exact ε-Pareto set, i.e., the modified algorithm yields the same cardinality guarantee as Algorithm 2. Since it requires a routine for Restrict^2_δ in addition to a routine for $\text{DualRestrict}^1_\delta$, the number of solved subproblems might be in the order of $\frac{M^2}{\varepsilon}$ (if this routine for Restrict^2_δ consists of simply applying the reduction to $\text{DualRestrict}^1_\delta$), which is by a factor of M larger than the number of solved subproblems in Algorithm 2.

For some common optimization problems such as biobjective ShortestPath, however, the best known algorithms for Restrict_δ and for $\text{DualRestrict}_\delta$ have similar asymptotic running times [9,11,13]. For other optimization problems such as minimum-cost maximum matching, a routine for Restrict_δ is available (see, e.g., [10]), but no specific routine for $\text{DualRestrict}_\delta$ is known, i.e., we can only solve $\text{DualRestrict}_\delta$ via the reduction to Restrict_δ. In these cases, our modification of the algorithm by Diakonikolas and Yannakakis offers an output-sensitive bound on the running time. It solves $\mathcal{O}(|P^*|)$ many subproblems if both routines are available and $\mathcal{O}(M \cdot |P^*|)$ many subproblems if only a routine for Restrict_δ is available. Algorithm 2 solves $\mathcal{O}(\frac{M}{\varepsilon})$ and $\mathcal{O}(\frac{M^2}{\varepsilon})$ many subproblems, respectively, in these cases, which is equal to the a-priori bound for the modified algorithm, but might be much larger than the a-posteriori bound for some optimization problems.

The modified algorithm is stated in Algorithm 3. Its proof of correctness is similar to the proof for the original algorithm by Diakonikolas and Yannakakis [6] and is given in the appendix. The algorithm explores the objective space bottom-right to top-left. It uses δ such that $(1+\delta)^3 = 1+\varepsilon$ and starts by solving $\text{DualRestrict}^1_\delta(2^M)$ and $\text{Restrict}^2_\delta(2^M)$, i.e., it computes a solution having minimum f_1-value and a solution that $(1, 1+\varepsilon)$-approximates all solutions having minimum f_2-value. It adds the latter one to the output set and then computes a solution that is not yet $(1, 1+\varepsilon)$-approximated and is worse in f_2 by a factor of at most $1+\delta$ than a solution having minimum f_2-value among all solutions that are not yet $(1, 1+\varepsilon)$-approximated. This is accomplished by solving $\text{Restrict}^1_\delta(B_1)$ for B_1 being the f_1-value of the previously added solution decreased by 2^{-2M}. Thus, a lower bound for the f_2-values of all solutions that are not yet approximated is known up to a (multiplicative) accuracy of $1+\delta$. Now, a solution that $(1, 1+\varepsilon)$-approximates all solutions with f_2-values in between this lower bound and this lower bound multiplied by $(1+\varepsilon)$ is computed using $\text{DualRestrict}^1_\delta$

[3] The term *output-sensitive* is often used to describe running times of algorithms that depend on the (a priori) unknown output of the algorithm like, e.g., the cardinality of a returned set of solutions.

and is added to the output set. This procedure is repeated until the solution having minimum f_1-value computed in the beginning of the algorithm is $(1, 1+\varepsilon)$-approximated.

Algorithm 3: An alternative $(1, 1+\varepsilon)$-approximation for biobjective optimization problems.

input : an instance (X, f) of a biobjective minimization problem, $\varepsilon > 0$, an algorithm for $\textsc{DualRestrict}^1_\delta$, an algorithm for $\textsc{Restrict}^2_\delta$, where $\delta = \sqrt[3]{1+\varepsilon} - 1$
output: a one-exact ε-Pareto set for (X, f)

$\delta \leftarrow \sqrt[3]{1+\varepsilon} - 1$
if $\textsc{Restrict}^2_\delta(2^M) = \text{NO}$ **then** halt
$x^{\text{left}} \leftarrow \textsc{DualRestrict}^1_\delta(2^M)$
$\tilde{x}^{(1)} \leftarrow \textsc{Restrict}^2_\delta(2^M)$
$S_2^{(1)} \leftarrow (1+\delta) \cdot f_2(\tilde{x}^{(1)})$
$x^{(1)} \leftarrow \textsc{DualRestrict}^1_\delta(S_2^{(1)})$
$B_1^{(1)} \leftarrow f_1(x^{(1)}) - 2^{-2M}$
$Q \leftarrow \{x^{(1)}\}$
$i \leftarrow 1$
while $B_1^{(i)} \geq f_1(x^{left})$ **do**
  $\tilde{x}^{(i+1)} \leftarrow \textsc{Restrict}^2_\delta(B_1^{(i)})$
  $S_2^{(i+1)} \leftarrow \frac{1+\varepsilon}{1+\delta} \cdot \max\{S_2^{(i)}, \frac{1}{1+\delta} \cdot f_2(\tilde{x}^{(i+1)})\}$
  $x^{(i+1)} \leftarrow \textsc{DualRestrict}^1_\delta(S_2^{(i+1)})$
  $B_1^{(i+1)} \leftarrow f_1(x^{(i+1)}) - 2^{-2M}$
  $Q \leftarrow Q \cup \{x^{(i+1)}\}$
  $i \leftarrow i + 1$
return Q

Theorem 8 *Algorithm 3 computes a one-exact ε-Pareto set Q of cardinality $|Q| \leq 2 \cdot |P^*|$ solving $\mathcal{O}(|P^*|)$ many instances of* $\textsc{DualRestrict}^1_\delta$ *and* $\textsc{Restrict}^2_\delta$*, where* $(1+\delta)^3 = 1+\varepsilon$.

Similar to the proof of correctness for the original algorithm for computing ε-Pareto sets by Diakonikolas and Yannakakis [6], the proof of Theorem 8 is based on comparing the cardinality of the set computed by Algorithm 3 to the cardinality of a smallest one-exact ε-Pareto set. This smallest one-exact ε-Pareto set is assumed to be computed by a greedy procedure similar to the ones given by Diakonikolas and Yannakakis [6], Koltun and Papadimitriou [12], and Vassilvitskii and Yannakakis [16]. The greedy procedure is based on an (exact) routine for Constrained and is given as Algorithm 4 in the appendix. The fact that $\textsc{Constrained}^1$ and $\textsc{Constrained}^2$ are polynomially equivalent via the same reduction as for $\textsc{DualRestrict}^1_\delta$ and $\textsc{Restrict}^2_\delta$ yields the following corollary:

Corollary 1 *For a biobjective optimization problem, it is possible to compute a smallest one-exact ε-Pareto set in fully polynomial time if a polynomial-time algorithm for* $\textsc{Constrained}^1$ *or* $\textsc{Constrained}^2$ *is available.*

Note that Constrained can, in particular, be solved efficiently if all feasible solutions are given explicitly in the input of a biobjective optimization problem. Thus, in this case, a smallest one-exact ε-Pareto set can be found in fully polynomial time.

5.4 Impossibility result for three or more objectives

In contrast to the biobjective case, we now demonstrate that no polynomial-time generic algorithm based on DualRestrict$_\delta$ can produce a constant-factor approximation on the cardinality of a smallest one-exact ε-Pareto set for general optimization problems with more than two objective functions.

Theorem 9 *For any $\varepsilon > 0$ and any positive integer $n \in \mathbb{N}_+$, there does not exist an algorithm that computes a one-exact ε-Pareto set P such that $|P| < n \cdot |P^*|$ for every 3-objective minimization problem and generates feasible solutions only via solving* DualRestrict$_\delta$ *for values of δ such that $\frac{1}{\delta}$ is polynomial in the encoding length of the input.*

Proof Given $\varepsilon > 0$ and $n \in \mathbb{N}_+$, we construct two instances I_1 and I_2 with $I_1 = (\{x_0, x_1, \ldots, x_n\}, f)$ and $I_2 = (\{x_0, x_1, \ldots, x_n, x'_1, \ldots, x'_n\}, f)$, where

$$f(x_i) = \begin{pmatrix} f_1(x_0) + n - i \\ (1+\varepsilon)^{2i} \cdot f_2(x_0) \\ \frac{1}{1+\varepsilon} \cdot f_3(x_0) \end{pmatrix} \text{ and } f(x'_i) = \begin{pmatrix} f_1(x_0) + n - i \\ (1+\varepsilon)^{2i} \cdot f_2(x_0) \\ \frac{1}{1+\varepsilon} \cdot f_3(x_0) - 1 \end{pmatrix}, i = 1, \ldots, n.$$

Then the solution x_0 $(1, 1+\varepsilon, 1+\varepsilon)$-approximates x_i for $i = 1, \ldots, n$, but x_0 does not $(1, 1+\varepsilon, 1+\varepsilon)$-approximate x'_i for any i due to the f_3-values. Also, no two solutions from the set $\{x_1, \ldots, x_n, x'_1, \ldots, x'_n\}$ approximate each other except for, possibly, x_i and x'_i for $i = 1, \ldots, n$. Thus, the set $\{x_0\}$ is a one-exact ε-Pareto set in instance I_1, but any one-exact ε-Pareto set in instance I_2 consists of at least $n+1$ solutions. Hence, an algorithm that computes a one-exact ε-Pareto set P with $|P| < n \cdot |P^*|$ has to be able to distinguish between I_1 and I_2, i.e., detect the existence of at least one x'_i, using only DualRestrict$_\delta$.

Following a similar argument as in the proof of Theorem 5, one can show that, in order to distinguish between the instances I_1 and I_2, an algorithm has to solve DualRestrict$^1_\delta$ for some δ with $\frac{1}{\delta} > \frac{1}{1+\varepsilon} \cdot f_2(x_0) - 1$. Since $f_2(x_0)$ can be exponential in the encoding length of the input, the claim follows. □

6 Conclusion and future research

This article addresses the task of computing approximate Pareto sets for multiobjective optimization problems. In particular, we strive for such approximate Pareto sets that are exact in one objective function (without loss of generality the first one) and obtain an approximation guarantee of $1+\varepsilon$ in all other objectives. We show the existence of such $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-approximate Pareto sets of polynomial cardinality under mild assumptions on the considered multiobjective optimization problem. Our main results address the relation between such a so-called one-exact ε-Pareto set and an auxiliary problem, the so-called DualRestrict$_\delta$ problem. Interestingly, this auxiliary problem has been considered in the literature before, but its full potential has not been revealed so far. In fact, we prove equivalence of computing a $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$-approximate Pareto set in polynomial time and solving DualRestrict$_\delta$ in polynomial time. This result complements the seminal work of Papadimitriou and Yannakakis [14], who characterize the class of multiobjective optimization problems for which a $(1+\varepsilon, \ldots, 1+\varepsilon)$-approximate Pareto set is polynomial-time computable using the so-called gap problem. With respect to the approximation quality, one cannot hope for a polynomial-time computable approximate Pareto set that is exact in more than one objective function since this would imply the polynomial-time solvability of a related biobjective optimization

problem. In this sense, the factor of $(1, 1+\varepsilon, \ldots, 1+\varepsilon)$ obtained here is best possible. Additionally, we provide an algorithm that approximates the cardinality of a smallest one-exact ε-approximate Pareto set by a factor of 2 for biobjective problems and show that this factor is best possible. Finally, we demonstrate that, using DualRestrict$_\delta$, it is not possible to obtain any constant-factor approximation on the cardinality for problems with more than two objectives efficiently.

It should be pointed out that our work provides a general method for computing polynomial-cardinality one-exact ε-Pareto sets by using an algorithm for the DualRestrict$_\delta$ problem. If applied to multiobjective SpanningTree, our work provides the first polynomial-time algorithm for computing a one-exact ε-Pareto set and, thus, yields the best possible approximation guarantee for this problem. For multiobjective ShortestPath, our general algorithms have running times that are competitive with the running time of the specialized algorithm of [15]. For biobjective ShortestPath, our algorithm additionally provides a worst-case guarantee on the cardinality of the computed one-exact ε-Pareto set. Future research could focus on the design of additional problem-specific algorithms that compute one-exact ε-Pareto sets for certain multiobjective (combinatorial) optimization problems with faster running times than the general methods provided here.

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/.

Appendix

In Algorithm 4, we formally state an algorithm that computes a smallest one-exact ε-Pareto set if a subroutine for solving Constrained is given. We prove its correctness in Theorem 10. Finally, we present a proof of Theorem 8.

Theorem 10 *Algorithm 4 computes a smallest one-exact ε-Pareto set P^* by solving $\mathcal{O}(|P^*|)$ instances of* Constrained^1 *and of* Constrained^2.

Proof First, note that

$$\begin{aligned} B_1^{(i+1)} + 2^{-2M} &= f_1(x^{(i+1)}) = \text{OPT}_1(B_2^{(i+1)}) \leq \text{OPT}_1(f_2(\tilde{x}^{(i+1)})) \\ &\leq f_1(\tilde{x}^{(i+1)}) \leq B_1^{(i)} \end{aligned}$$

for $i = 1, 2, \ldots$, where all the steps follow immediately from the algorithm and the definition of $\text{OPT}_1(\cdot)$. Thus, the termination condition $B_1^{(i)} < f_1(x^{\text{left}})$ is fulfilled after finitely many iterations and Algorithm 4 returns a set P^* of finite cardinality. Moreover, we obtain the following statements:

(a) $x^{(1)}$ $(1, 1+\varepsilon)$-approximates any solution $x' \in X$ with $f_1(x') \geq f_1(x^{(1)})$. This is because in the f_2-component, we have

$$f_2(x^{(1)}) \leq B_2^{(1)} = (1+\varepsilon) \cdot f_2(\tilde{x}^{(1)}) = (1+\varepsilon) \cdot \text{OPT}_2(2^M) \leq (1+\varepsilon) \cdot f_2(x').$$

Algorithm 4: Greedy algorithm for computing a smallest one-exact ε-Pareto set for biobjective optimization problems.

input : an instance (X, f) of a biobjective minimization problem, $\varepsilon > 0$, an algorithm for $\textsc{Constrained}^1$, an algorithm for $\textsc{Constrained}^2$
output: a one-exact ε-Pareto set for (X, f)

if $\textsc{Constrained}^2(2^M) = \text{NO}$ **then** halt
$x^{\text{left}} \leftarrow \textsc{Constrained}^1(2^M)$
$\tilde{x}^{(1)} \leftarrow \textsc{Constrained}^2(2^M)$
$B_2^{(1)} \leftarrow (1+\varepsilon) \cdot f_2(\tilde{x}^{(1)})$
$x^{(1)} \leftarrow \textsc{Constrained}^1(B_2^{(1)})$
$B_1^{(1)} \leftarrow f_1(x^{(1)}) - 2^{-2M}$
$P^* \leftarrow \{x^{(1)}\}$
$i \leftarrow 1$
while $B_1^{(i)} \geq f_1(x^{left})$ **do**
  $\tilde{x}^{(i+1)} \leftarrow \textsc{Constrained}^2(B_1^{(i)})$
  $B_2^{(i+1)} \leftarrow (1+\varepsilon) \cdot f_2(\tilde{x}^{(i+1)})$
  $x^{(i+1)} \leftarrow \textsc{Constrained}^1(B_2^{(i+1)})$
  $B_1^{(i+1)} \leftarrow f_1(x^{(i+1)}) - 2^{-2M}$
  $P^* \leftarrow P^* \cup \{x^{(i+1)}\}$
  $i \leftarrow i + 1$
return P^*

(b) For $i = 2, \ldots, |P^*|$, the solution $x^{(i)}$ $(1, 1+\varepsilon)$-approximates any solution $x' \in X$ with $f_1(x^{(i-1)}) > f_1(x') \geq f_1(x^{(i)})$ because in the f_2-component, we have

$$\begin{aligned} f_2(x^{(i)}) \leq B_2^{(i)} &= (1+\varepsilon) \cdot f_2(\tilde{x}^{(i)}) = (1+\varepsilon) \cdot \textsc{opt}_2(B_1^{(i-1)}) \\ &\leq (1+\varepsilon) \cdot \textsc{opt}_2(f_1(x^{(i-1)})) \leq (1+\varepsilon) \cdot f_2(x'). \end{aligned}$$

(c) There are no solutions $x' \in X$ with $f_1(x') < f_1(x^{(|P^*|)})$ since otherwise we would have

$$f_1(x') \leq B_1^{(|P^*|)} < f_1(x^{\text{left}}) = \textsc{opt}_1(2^M),$$

where the strict inequality holds due to the termination condition of the algorithm.

Statements (a)–(c) imply that the set P^* computed by Algorithm 4 is a one-exact ε-Pareto set.

We now show via induction that, for all $k \in \{1, \ldots, |P^*|\}$, there exists a smallest one-exact ε-Pareto set P_k^* such that $\{x^{(1)}, \ldots, x^{(k)}\} \subseteq P_k^*$. This fact for $k = |P^*|$ then yields $|P^*| \leq |P_{|P^*|}^*|$, which completes the proof.

In order to prove the claim for $k = 1$, consider a smallest one-exact ε-Pareto set P_0^*. Let $x_1 \in P_0^*$ be a solution that $(1, 1+\varepsilon)$-approximates $\tilde{x}^{(1)}$ and let $x' \in X$ be an arbitrary solution $(1, 1+\varepsilon)$-approximated by x_1. In the second component, we have $f_2(x_1) \leq (1+\varepsilon) \cdot f_2(\tilde{x}^{(1)}) = B_2^{(1)}$, so, in the first component, we have

$$f_1(x^{(1)}) = \textsc{opt}_1(B_2^{(1)}) \leq f_1(x_1) \leq f_1(x').$$

Now, (a) implies that $x^{(1)}$ also $(1, 1+\varepsilon)$-approximates x' and, thus, x_1 can be replaced by $x^{(1)}$ in P_0^*.

For the induction step $k \to k+1$, let P_k^* be a smallest one-exact ε-Pareto set with $\{x^{(1)}, \ldots, x^{(k)}\} \subseteq P_k^*$. Let $x_{k+1} \in P_k^*$ be a solution that $(1, 1+\varepsilon)$-approximates $\tilde{x}^{(k+1)}$ and note that $f_1(x_{k+1}) \leq f_1(\tilde{x}^{(k+1)}) \leq B_1^{(k)} < f_1(x^{(k)})$, so we must have $x_{k+1} \in P_k^* \setminus \{x^{(1)}, \ldots, x^{(k)}\}$. Now, let $x' \in X$ be an arbitrary solution that is $(1, 1+\varepsilon)$-approximated by x_{k+1} but not by any other solution in the set P_k^*. In the first component, we then have

$$f_1(x^{(k+1)}) = \text{OPT}_1(B_2^{(k+1)}) \leq f_1(x_{k+1}) \leq f_1(x') < f_1(x^{(k)}),$$

where the strict inequality holds due to (a) and (b) and because x' is not $(1, 1+\varepsilon)$-approximated by $x^{(1)}, \ldots, x^{(k)}$. Now, (b) implies that $x^{(k+1)}$ also $(1, 1+\varepsilon)$-approximates x' and, thus, x_{k+1} can be replaced by $x^{(k+1)}$ in P_k^*. □

We now present the proof of Theorem 8.

Proof of Theorem 8 The proof is similar to the proof of the non-modified algorithm by Diakonikolas and Yannakakis [6] and to the proof of Theorem 10. In fact, the first part of the proof, where it is shown that Algorithm 3 correctly computes a one-exact ε-Pareto set Q follows exactly the same steps as the proof of Theorem 10, so we omit this part here.

We prove the bound on the cardinality of Q by comparing Q to a smallest one-exact ε-Pareto set $P^* = \{x_*^{(1)}, \ldots, x_*^{(|P^*|)}\}$ computed by Algorithm 4.

We prove that the following statement holds for all $k \in \mathbb{N}$ by induction on k (which immediately implies the claim): If $|Q| \geq 2k-1$, then $|P^*| \geq k$, and if $|Q| \geq 2k$, then $f_1(x_*^{(k)}) \geq f_1(x^{(2k)})$.

For $k = 1$, it suffices to show that, if $|Q| \geq 2$, we have $f_1(x_*^{(1)}) \geq f_1(x^{(2)})$. In order to see this, note that

$$f_2(x_*^{(1)}) \leq (1+\varepsilon) \cdot \text{OPT}_2(2^M) \leq (1+\varepsilon) \cdot f_2(\tilde{x}^{(1)}) = \frac{1+\varepsilon}{1+\delta} \cdot S_2^{(1)} \leq S_2^{(2)}.$$

This implies that $f_1(x_*^{(1)}) \geq \text{OPT}_1(S_2^{(2)}) \geq f_1(x^{(2)})$.

Now suppose that, for some $k \geq 2$, we have $|Q| \geq 2k-2$ and $f_1(x_*^{(k-1)}) \geq f_1(x^{(2k-2)})$. We first show that if $|Q| \geq 2k-1$, then $|P^*| \geq k$. Recall that the sequence $(f_1(x_*^{(1)}), \ldots, f_1(x_*^{(|P^*|)}))$ is strictly decreasing. Since the solution x^{left} must be $(1, 1+\varepsilon)$-approximated by some solution in P^*, we must have $f_1(x_*^{(|P^*|)}) \leq f_1(x^{\text{left}})$, but for $x_*^{(k-1)}$, we have

$$f_1(x_*^{(k-1)}) \geq f_1(x^{(2k-2)}) > B_1^{(2k-2)} \geq f_1(x^{\text{left}}),$$

so $|P^*| > k-1$.

Finally, we show that if $|Q| \geq 2k$, then $f_1(x_*^{(k)}) \geq f_1(x^{(2k)})$. Note that we have $f_1(\tilde{x}^{(2k-1)}) \leq B_1^{(2k-2)} < f_1(x^{(2k-2)}) \leq f_1(x_*^{(k-1)})$ due to the induction hypothesis, so $\tilde{x}^{(2k-1)}$ is not $(1, 1+\varepsilon)$-approximated by $x_*^{(1)}, \ldots, x_*^{(k-1)}$. Let $x_*^{(i)} \in P^*$ be a solution that $(1, 1+\varepsilon)$-approximates $\tilde{x}^{(2k-1)}$. In the second component, we then have

$$f_2(x_*^{(i)}) \leq (1+\varepsilon) \cdot f_2(\tilde{x}^{(2k-1)}) \leq \frac{1+\varepsilon}{1+\delta} \cdot S_2^{(2k-1)} \leq S_2^{(2k)}.$$

The above argument implies $i \geq k$, so

$$f_1(x_*^{(k)}) \geq f_1(x_*^{(i)}) \geq \text{OPT}_1(S_2^{(2k)}) \geq f_1(x^{(2k)}),$$

which finishes the proof. □

References

1. Angel, E., Bampis, E., Kononov, A.: A FPTAS for approximating the unrelated parallel machines scheduling problem with costs. In: Proceedings of the 9th Annual European Symposium on Algorithms (ESA), LNCS, vol. 2161, pp. 194–205 (2001)
2. Angel, E., Bampis, E., Kononov, A.: On the approximate tradeoff for bicriteria batching and parallel machine scheduling problems. Theor. Comput. Sci. **306**, 319–338 (2003)
3. 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)
4. Bökler, F.: Output-sensitive complexity of multiobjective combinatorial optimization with an application to the multiobjective shortest path problem. Ph.D. thesis, Technical University of Dortmund (2018)
5. Bökler, F., Ehrgott, M., Morris, C., Mutzel, P.: Output-sensitive complexity of multiobjective combinatorial optimization. J. Multi-Criteria Decis. Anal. **24**(1–2), 25–36 (2017)
6. Diakonikolas, I., Yannakakis, M.: Small approximate Pareto sets for biobjective shortest paths and other problems. SIAM J. Comput. **39**(4), 1340–1371 (2009)
7. 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)
8. Ehrgott, M.: Multicriteria Optimization. Springer, Berlin (2005)
9. Ergun, F., Sinha, R., Zhang, L.: An improved FPTAS for restricted shortest path. Inf. Process. Lett. **83**(5), 287–291 (2002)
10. Grandoni, F., Ravi, R., Singh, M., Zenklusen, R.: New approaches to multi-objective optimization. Math. Program. **146**(1), 525–554 (2014)
11. Horváth, M., Kis, T.: Multi-criteria approximation schemes for the resource constrained shortest path problem. Optim. Lett. **12**(3), 475–483 (2018)
12. Koltun, V., Papadimitriou, C.: Approximately dominating representatives. Theor. Comput. Sci. **371**(3), 148–154 (2007)
13. Lorenz, D.H., Raz, D.: A simple efficient approximation scheme for the restricted shortest path problem. Oper. Res. Lett. **28**(5), 213–219 (2001)
14. 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)
15. Tsaggouris, G., Zaroliagis, C.: Multiobjective optimization: improved FPTAS for shortest paths and nonlinear objectives with applications. Theory Comput. Syst. **45**(1), 162–186 (2009)
16. 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

Arne Herzel[1,3] · Cristina Bazgan[2] · Stefan Ruzika[1] · Clemens Thielen[3] · Daniel Vanderpooten[2]

Cristina Bazgan
bazgan@lamsade.dauphine.fr

Stefan Ruzika
ruzika@mathematik.uni-kl.de

Clemens Thielen
clemens.thielen@tum.de

Daniel Vanderpooten
daniel.vanderpooten@lamsade.dauphine.fr

1 Department of Mathematics, University of Kaiserslautern, Paul-Ehrlich-Str. 14, 67663 Kaiserslautern, Germany

[2] PSL Research University, CNRS, LAMSADE, Université Paris-Dauphine, 75016 Paris, France

[3] TUM Campus Straubing for Biotechnology and Sustainability, Technical University of Munich, Am Essigberg 3, 94315 Straubing, Germany