Computers & Operations Research 145 (2022) 105857

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

What if we increase the number of objectives? Theoretical and empirical implications for many-objective combinatorial optimization

Richard Allmendinger [a,*], Andrzej Jaszkiewicz [b], Arnaud Liefooghe [c], Christiane Tammer [d]

[a] *The University of Manchester, Alliance Manchester Business School, Booth Street West, Manchester M15 6PB, UK*
[b] *Poznan University of Technology, Institute of Computing Science, Piotrowo 2, 60-965 Poznań, Poland*
[c] *Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France*
[d] *Martin-Luther-Universität Halle-Wittenberg, Institut für Mathematik, 06099 Halle, Germany*

ARTICLE INFO

Keywords:
Multi- and many-objective optimization
Problem characteristics
Complexity of procedures and algorithms
Survey
Theoretical and empirical analysis

ABSTRACT

The difficulty of solving a multi-objective optimization problem is impacted by the number of objectives to be optimized. The presence of many objectives typically introduces a number of challenges that affect the choice/design of optimization algorithms. This paper investigates the drivers of these challenges from two angles: (i) the influence of the number of objectives on problem characteristics and (ii) the practical behavior of commonly used procedures and algorithms for coping with many objectives. In addition to reviewing various drivers, the paper makes theoretical contributions by quantifying some drivers and/or verifying these drivers empirically by carrying out experiments on multi-objective combinatorial optimization problems (multi-objective NK-landscapes). We then make use of our theoretical and empirical findings to derive practical recommendations to support algorithm design. Finally, we discuss remaining theoretical gaps and opportunities for future research in the area of multi- and many-objective optimization.

1. Introduction

Multi-objective optimization (MO) (Deb, 2001; Miettinen, 2012) is the area looking at the development and application of algorithms to problems with multiple conflicting objectives; problems with more than three objectives have also been termed as many-objective problems (Kollat et al., 2011) and are less studied. In the absence of any user preferences about desired ideal solutions, the goal of MO algorithms (MOAs) is not to identify a single optimal solution but to approximate (or compute exactly if possible) the set of best trade-off solutions to a problem, also known as the Pareto (optimal) set. We say that a solution Pareto dominates, or simply dominates, another solution if it is not worse in any objective and if it is strictly better in at least one objective. Solutions in the Pareto set are non-dominated by any other solution from the feasible solution space.

The traditional approach to tackle a MO problem is to convert it into a single-objective problem using a scalarizing function (Chankong and Haimes, 1983; Pascoletti and Serafini, 1984; Eichfelder, 2008), and then solve the problem repeatedly using different 'configurations' of the scalarizing function (e.g. different weights). A commonly used scalarizing method is to combine the objectives using a weighted sum, and then alter the weights to discover one solution per algorithmic run. However, in the non-convex case, non-linear scalarization methods are more useful, given that linear scalarization methods can generate only a subset of Pareto optimal solutions, known as supported solutions (Miettinen, 2012). An alternative approach is to use a population-based approach, such as a multi-objective evolutionary algorithm (MOEA), to evolve several solutions (a population) in one algorithmic run with the hope to also discover multiple Pareto optimal solutions in one go. MOEAs have proven to be an efficient approach, especially for problems with two and three objectives (Coello et al., 2007; Deb, 2001). Traditional MOEAs rely on the Pareto dominance concept combined with diversity maintenance mechanisms to drive the search (e.g. NSGA-II Deb et al., 2002). Two further prominent concepts are indicator-based and decomposition-based methods. The former concept replaces the objectives with a unary set performance metric (indicator) and then optimizes this metric (e.g. AGA Knowles and Corne, 2003 and IBEA Zitzler and Künzli, 2004). A commonly used indicator is the hypervolume indicator (Zitzler et al., 2003), which measures the volume of the objective space that is dominated by the image of the Pareto set (also known as the Pareto front) and bounded by a reference point. Decomposition-based approaches (e.g. MOEA/D Zhang and Li, 2007) decompose a MO problem into several single-objective problems using a scalarizing function with different weights. Each solution in the population is then dedicated to optimize one scalarizing function.

[*] Corresponding author.
E-mail address: richard.allmendinger@manchester.ac.uk (R. Allmendinger).

https://doi.org/10.1016/j.cor.2022.105857
Received 5 October 2020; Received in revised form 25 March 2022; Accepted 21 April 2022
Available online 14 May 2022

From the literature, it is apparent that solving MO problems is known to be difficult for some problem classes; moreover, we know from the No Free Lunch theorem (Wolpert and Macready, 1997) that there is no single best approach in the general case. The computational difficulty grows with an increase in the number of objectives affecting different problem characteristics, such as the number of Pareto optimal solutions, distances among solutions, and likelihood of objectives varying in complexity and evaluation times. Furthermore, it is anticipated that algorithms and procedures designed for MO problems will face difficulties as the number of objectives increases. For example, the complexity for routines such as dominance tests and updating of the Pareto archive affects Pareto dominance-based MOEAs, creating evenly distributed weight vectors impacts decomposition-based MOEAs and scalarizing methods, and computing and approximating the hypervolume can become an issue for indicator-based MOEAs and performance validation. An overview of recent developments in the area of many-objective optimization can be found, for example, in Ishibuchi et al. (2008), Aguirre (2013) and Chand and Wagner (2015), and empirical work on the efficiency of MOEAs for many-objective optimization problems can be found, for example, in Purshouse and Fleming (2003), Hughes (2005) and Wagner et al. (2007).

We make several contributions to support the community in gaining a better understanding about the effect of increasing the number of objectives on problem characteristics and the complexity of MO procedures and algorithms:

- We adopt a holistic approach linking theory with empirical analysis, and then we highlight how our results translate into practice.
- We theoretically investigate the key drivers attributing to an increase in computational and algorithmic challenges. In particular, we derive probabilities for a solution to be non-dominated, and propose a general formulation for scalarizing functions.
- We conduct empirical experiments on multi-objective NK-landscapes (Aguirre and Tanaka, 2007; Verel et al., 2013) to back up our theoretical contributions and verify various published theoretical results (as summarized in Table 1).
- We derive practical recommendations from our analysis to support algorithm design.

Let us highlight that our focus is on understanding the impact of the number of objectives on different problem characteristics and MO procedures including exact scalarization-based methods and population-based heuristics. The review and evaluation of individual MOEAs is out of scope of this paper, and can be found, for example, in Ishibuchi et al. (2008) and Chand and Wagner (2015). However, as pointed out above, we will use our theoretical and empirical findings to provide recommendations for MOEAs of different type.

This paper is organized as follows. The next section defines key MO concepts and motivates the choice of the test problem (NK-landscapes) used for empirical evaluation. Section 3 investigates the influence of the number of objectives on different problem characteristics, while Section 4 investigates the complexity of commonly used procedures and algorithms for coping with multiple objectives. Section 5 presents a summary of our theoretical and empirical findings, and then uses these to make recommendations for algorithmic setup choices for MOEAs, before finally discussing directions for future research.

2. Background

This section provides the relevant formal definitions to understand and avoid ambiguity of the various MO concepts, which will be used in the subsequent sections. The section will also provide details of multi-objective NK-landscapes, the model we use to empirically validate our theoretical contributions and/or existing theoretical results.

2.1. Basic definitions

Definition 2.1 (*Multi-Objective Optimization (MO) Problem*). The general formulation of a MO problem is to "maximize" $\mathbf{f}(\mathbf{x})$ subject to $\mathbf{x} \in X$, where $\mathbf{x} = (x_1, \dots, x_j, \dots, x_n)$ is a candidate solution vector (or simply solution) consisting of n design (or decision) variables, $X \subset \mathbb{R}^n$ is the search domain, and $\mathbf{f} = (f_1, \dots, f_i, \dots, f_m)$ is a vector objective function $\mathbf{f} : X \rightarrow \mathbb{R}^m$ mapping solutions to a m dimensional objective space. The term "maximize" is written in quotes to indicate that in general there is no single solution that maximizes all objectives simultaneously, and a further definition is needed to define an ordering on candidate solutions (see below).

Definition 2.2 (*Pareto Dominance*). Consider two solutions $\mathbf{x}^1 \in X$ and $\mathbf{x}^2 \in X$. We say that $\mathbf{x}^1$ is dominated by $\mathbf{x}^2$, also written as $\mathbf{x}^1 \prec \mathbf{x}^2$, if and only if $\exists i$ such that $f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2)$ and $\forall j, f_j(\mathbf{x}^1) \leq f_j(\mathbf{x}^2)$. This relation is also sometimes called strict Pareto dominance in contrast to the weak Pareto dominance defined below.

Definition 2.3 (*Weak Pareto Dominance*). Consider two solutions $\mathbf{x}^1 \in X$ and $\mathbf{x}^2 \in X$. We say that $\mathbf{x}^1$ is weakly dominated by $\mathbf{x}^2$, also written as $\mathbf{x}^1 \preceq \mathbf{x}^2$, if and only if $\forall j, f_j(\mathbf{x}^1) \leq f_j(\mathbf{x}^2)$.

Definition 2.4 (*Pareto Optimal*). A solution $\mathbf{x}^1 \in X$ is called Pareto optimal if there does not exist a solution $\mathbf{x}^2 \in X$ that dominates it.

Definition 2.5 (*Pareto Set*). The set of all Pareto optimal solutions is said to form the Pareto set.

Definition 2.6 (*Pareto Front*). The image of the Pareto set in the objective space is known as the Pareto front.

Definition 2.7 (*Hypervolume Indicator*). Given a set of points in the objective space $S \subset \mathbb{R}^m$ and a reference point $r \in \mathbb{R}^m$, the hypervolume indicator of S is the measure of the region weakly dominated by S and weakly dominating r, i.e.:

$$H(S, r) = \Lambda(q \in \mathbb{R}^m \mid r \preceq q \ \wedge \ \exists p \in S : q \preceq p)$$

where $\Lambda(.)$ denotes the Lebesgue measure. Alternatively, it may be interpreted as the measure of the union of boxes:

$$H(S) = \Lambda\Big(\bigcup_{p \in S, r \preceq p} [r, p] \Big)$$

where $[r, p] = \{q \in \mathbb{R}^m \mid q \preceq p \ \wedge \ r \preceq q\}$ denotes the hypercuboid delimited by p and r. The advantage of hypervolume indicator is its compliance with the Pareto dominance relation (Zitzler et al., 2003).

Definition 2.8 (*Pareto Archive*). A Pareto archive is a set of mutually non-dominated solutions, i.e.:

$$A = \{\mathbf{x} \in X \mid \forall \mathbf{x}^1 \in A, \mathbf{x}^2 \in A \ : \ \mathbf{x}^2 \nprec \mathbf{x}^1 \ \wedge \ \mathbf{x}^1 \nprec \mathbf{x}^2\}.$$

In the context of multi-objective algorithms, a Pareto archive is used to store potentially Pareto optimal solutions, i.e. solutions that are not dominated by any solution generated so far. The Pareto archive may be unbounded, or bounded in cardinality, i.e. it may contain only a limited number of N solutions.

2.2. Multi-objective NK-landscapes

Multi-objective NK-landscapes (Aguirre and Tanaka, 2007; Verel et al., 2013) constitute a synthetic problem-independent model used for constructing multi-objective combinatorial optimization problems. They extend single-objective NK-landscapes (Kauffman, 1993), describing and generalizing a large family of unconstrained binary optimization problems (Heckendorn and Whitley, 1997). Importantly, a multi-objective NK-landscape instance can be constructed with any number of objectives, making it attractive for the purpose of our study.

Table 1
Overview of the topics to which this paper is making either a theoretical or empirical contribution (using NK-landscapes), and the section, equation(s) and/or figure(s) presenting the actual contribution. Sections not listed in the table perform a mini review and are included in the paper to put the work in context.

Topic	Presented in	Theoretical contribution	Empirical contribution
Number of Pareto optimal solutions	Section 3.1		Fig. 1
Probability for a solution to be non-dominated	Section 3.2	Eq. (1)	Figs. 2–5
Probability of heterogeneous objectives	Section 3.4		Fig. 6
Distances between solutions	Section 3.5		Fig. 7
Dominance test and updating the Pareto archive	Section 4.1		Figs. 9–11
Computing and approximating hypervolume	Section 4.2		Figs. 12–14
Impact on scalarization methods	Section 4.3	Eq. (A.10)	
Distances between weight vectors	Section 4.4		Fig. 15

In particular, candidate solutions are binary strings of size n. The objective function vector $\mathbf{f}$ is defined as $\mathbf{f}: \{0,1\}^n \rightarrow [0,1]^m$ such that each objective f_i is to be maximized. As in the single-objective case, the objective value $f_i(\mathbf{x})$ of a solution $\mathbf{x}$ is the average value of the individual contributions associated with each design variable x_j. Given objective f_i, $i \in \{1,\dots,m\}$, and each variable x_j, $j \in \{1,\dots,n\}$, a component function $f_{ij}: \{0,1\}^{k+1} \rightarrow [0,1]$ assigns a real-valued contribution for every combination of x_j and its k *epistatic interactions* $\left\{x_{j_1},\dots,x_{j_k}\right\}$. These f_{ij}-values are uniformly distributed in $[0,1]$. Thus, the individual contribution of a variable x_j depends on its value and on the values of $k < n$ variables $\left\{x_{j_1},\dots,x_{j_k}\right\}$ other than x_j. The problem can be formalized as follows:

$$\begin{array}{lll} \max & f_i(x) = \dfrac{1}{n}\displaystyle\sum_{j=1}^{n} f_{ij}(x_j, x_{j_1},\dots,x_{j_k}) & i \in \{1,\dots,m\} \\ \text{s.t.} & x_j \in \{0,1\} & j \in \{1,\dots,n\} \end{array}$$

The epistatic interactions, i.e. the k variables that influence the contribution of x_j, are typically set uniformly at random among the $(n-1)$ variables other than x_j, following the random neighborhood model from Kauffman (1993). By increasing the number of epistatic interactions k from 0 to $(n-1)$, problem instances can be gradually tuned from smooth to rugged. Interestingly, multi-objective NK-landscapes exhibit different characteristics and different degrees of difficulty for multi-objective optimization methods[1] (Daolio et al., 2017; Liefooghe et al., 2020). In the following two sections, if not otherwise stated we have considered 30 randomly-generated instances of multi-objective NK-landscapes with $n = 10$ decision variables, and $k = 0$ (i.e. linear problems), but, of course, varied the number of objectives m as the impact of m is the focus of this study.

3. Effect of the number of objectives on problem characteristics

In this section, we study the influence of the number of objectives on different problem characteristics as outlined in Table 1.

3.1. Number of Pareto optimal solutions

In the combinatorial case, the number of Pareto optimal solutions is known to grow exponentially with the number of objectives in the worst case, that is $\mathcal{O}(c^{m-1})$, where c is a constant value (Bazgan et al., 2013). Furthermore, as shown by Bazgan et al. (2013), this bound is tight for many classical multi-objective combinatorial optimization problems, such as selection, knapsack, shortest path, spanning tree, traveling salesperson, and s–t cut problems. Obviously, the number of Pareto optimal solutions is also bounded by the size of the whole feasible set.

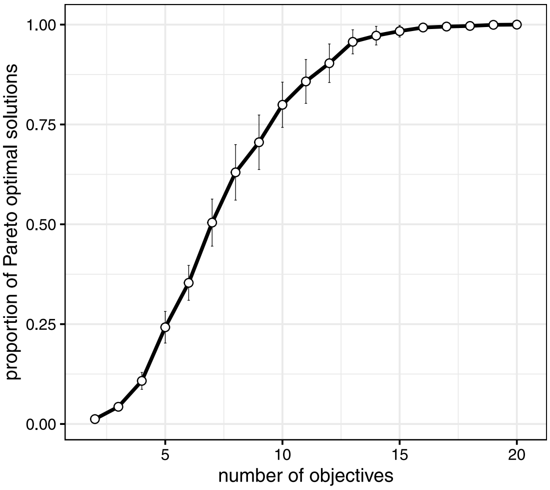

Fig. 1. Proportional number of Pareto optimal solutions with respect to the number of objectives m for multi-objective NK-landscapes.

Verel et al. (2013) experimentally investigate the number of Pareto optimal solutions for multi-objective NK-landscapes with 2, 3 and 5 objectives. They report that it grows exponentially with the problem size, the number of objectives and the degree of conflict among the objectives, and that it slightly decreases with the number of variable interactions. Given the aim of the current paper, we focus on the number of objectives by considering multi-objective NK-landscapes with 2 to 20 objectives. In Fig. 1, we report the proportion of Pareto optimal solutions in the solution space with respect to the number of objectives. We see that less than 5% of solutions are Pareto optimal for bi-objective problems ($m = 2$), whereas this proportion grows to about 50% for $m = 7$ objectives. For $m = 20$ objectives, more than 99% of solutions are Pareto optimal for all considered instances.

3.2. Probability for a solution to be non-dominated

With a growing number of objectives, the dominance relation becomes less discriminative (Aguirre, 2013). Let us consider the comparison between two arbitrary solutions $\mathbf{x}^1$ and $\mathbf{x}^2$ on m objectives. Assume that the probability of equal objective values can be neglected, and that the comparison with respect to each objective is independent. For each objective there is a $1/2$ probability that $\mathbf{x}^1$ has a better value for this objective than $\mathbf{x}^2$, and the same probability applies for the opposite situation. Note that we do not assume any particular distribution of

[1] The source code of the multi-objective NK-landscapes generator is available at the following URL: http://mocobench.sf.net.

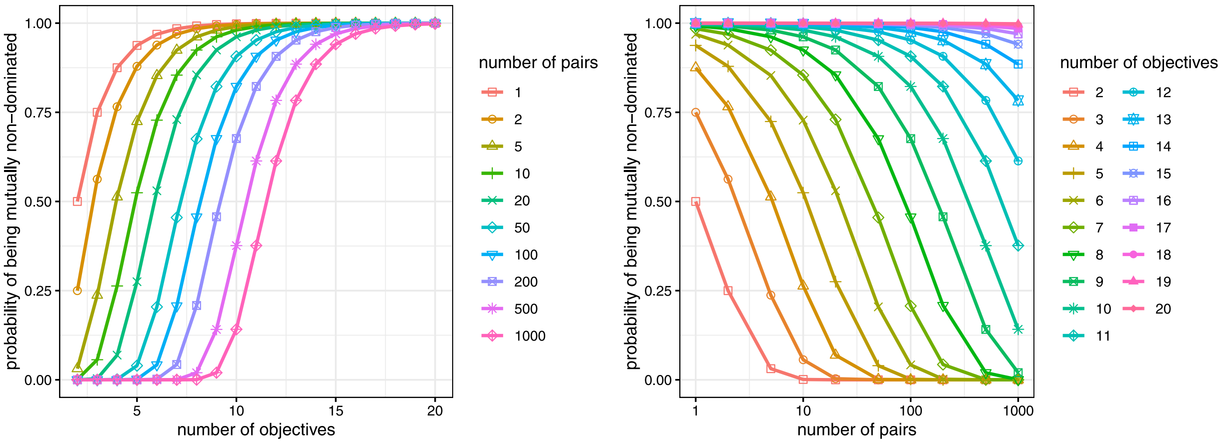

Fig. 2. Probability that all μ randomly selected pairs of solutions are mutually non-dominated from the theoretical model given in Eq. (1), with respect to the number of objectives m (left), and to the number of pairs μ (right).

objective values. We just assume that the results of comparisons according to particular objectives are independent random variables. As such, given a problem with m objectives, the probability that one solution dominates the other one is $1/2^{(m-1)}$. Thus, as m increases, it becomes more likely that two arbitrary solutions are mutually non-dominated. If the objectives are positively correlated, this probability increases, and if they are negatively correlated this probability decreases (Verel et al., 2013).

As a consequence of the reduced discriminative power of the dominance relation, the probability that a given solution is Pareto optimal increases with the number of objectives. To the best of our knowledge, there is no exact formula for the probability of a solution to be Pareto optimal within a given population, and such probability would depend on the distribution of solutions in the population. However, to get some intuition about the impact of the number of objectives on this, we derive the formula for the probability that all μ randomly selected pairs of solutions are mutually non-dominated:

$$\prod_{i=1}^{\mu}\left(1-\frac{1}{2^{m-1}}\right) = \left(1-\frac{1}{2^{m-1}}\right)^{\mu} \tag{1}$$

This theoretical probability is reported in Fig. 2 for different number of objectives, m, and number of pairs, μ. As such, the probability that all μ randomly selected pairs of solutions are mutually non-dominated increases with the number of objectives, and decreases with the number of pairs. It becomes very likely that all pairs are mutually non-dominated for problems with $m > 15$ objectives, even for large numbers of pairs. In Fig. 3, we measure this proportion for multi-objective NK-landscapes using $m \in \{2, 3, \ldots, 20\}$. In particular, we perform 30 independent samples per instance for each setting. Comparing Fig. 2 with Fig. 3, we observe that the theoretical model fits the experimental data well, although there are some small but significant differences. Through an additional analysis we found that this is caused by small positive or negative correlations between objectives resulting from random (diss)similarities of instance parameters. Such correlations modify the probabilities that two randomly selected solutions are non-dominated w.r.t. the assumed independent objectives model. Since Eq. (1) is exponential, even very small differences of the probabilities that two randomly selected solutions are non-dominated may results in noticeable differences in the probability that all pairs are non-dominated.

Although the probability for a solution to be non-dominated in random populations might be difficult to derive theoretically due to the dependencies between solutions, we report in Fig. 4 empirical results on the proportional number of times a given solution is *not* dominated by any of μ other solutions, all drawn at random. We consider the same set of multi-objective NK-landscapes. The probability for a solution to be non-dominated grows with the number of objectives, but the increase is steeper than that of Fig. 3. It also reduces with the population size, although its effect is lower than that of the number of objectives. For 2 objectives, it ranges from about 70% when a single pair of solutions is considered, to less than 2% when the solution is compared against 1 000 others. By contrast, for problems with more than 12 objectives, there is over 90% chance that the solution is not dominated by any other, independently of the population size.

Additionally, we report in Fig. 5 the proportional number of non-dominated solutions in populations containing μ random solutions. The curves follow a similar trend than that of Fig. 4. For small populations ($\mu = 10$), there is about 30% of non-dominated solutions for 2-objective problems, 50% for 3-objective problems, to more than 70% for problems with 4 objectives and more, and even more than 90% for problems with 6 objectives and more. For larger populations, this proportion decreases but remains higher than 90% for problems with 12 objectives or more, independently of the population size. Multi-objective selection methods for ranking solutions from a population of mutually non-dominated solutions are discussed and compared by Corne and Knowles (2007).

3.3. Number of preference parameters

In numerous multi-objective optimization methods, the decision maker (DM) is expected to express his/her preferences, e.g. in the form of weighting coefficients or reference levels (aspiration levels/goals) specified for each objective (Miettinen, 2012). The number of such preference parameters grows just linearly with the number of objectives. However, for certain methods, such as AHP (Saaty, 1987), preference parameters are expressed with respect to each pair of objectives; in this case, their number grows quadratically.

3.4. Probability of having heterogeneous objectives

As the number of objectives increases and one is faced with a many-objective problem, it becomes more likely that the objectives are heterogeneous meaning they differ in, for example, complexity (e.g. linear vs non-linear, unimodal vs multimodal), evaluation efforts in terms of time, costs, or resources, available information (expensive black box vs given closed-form function), or determinism (stochastic vs deterministic).

Examples of problems with heterogeneous objectives include real-world problems where some objectives must be evaluated using time-consuming physical experiments in the laboratory, while other objectives are evaluated using relatively quicker calculations on the computer. Problems that fall into this category can be found, for instance, in

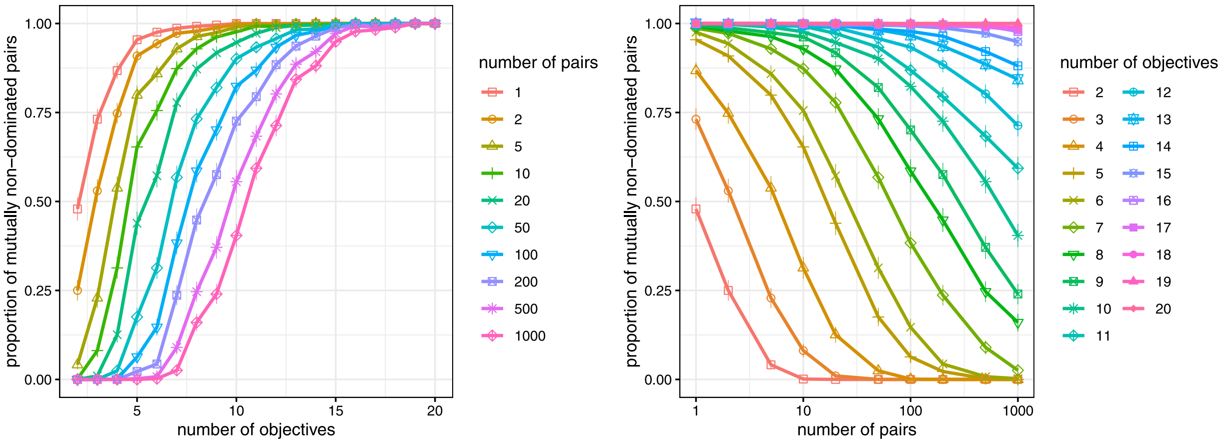

Fig. 3. Proportion from μ randomly selected pairs of solutions that are mutually non-dominated for multi-objective NK-landscapes, with respect to the number of objectives m (left), and to the number of pairs μ (right).

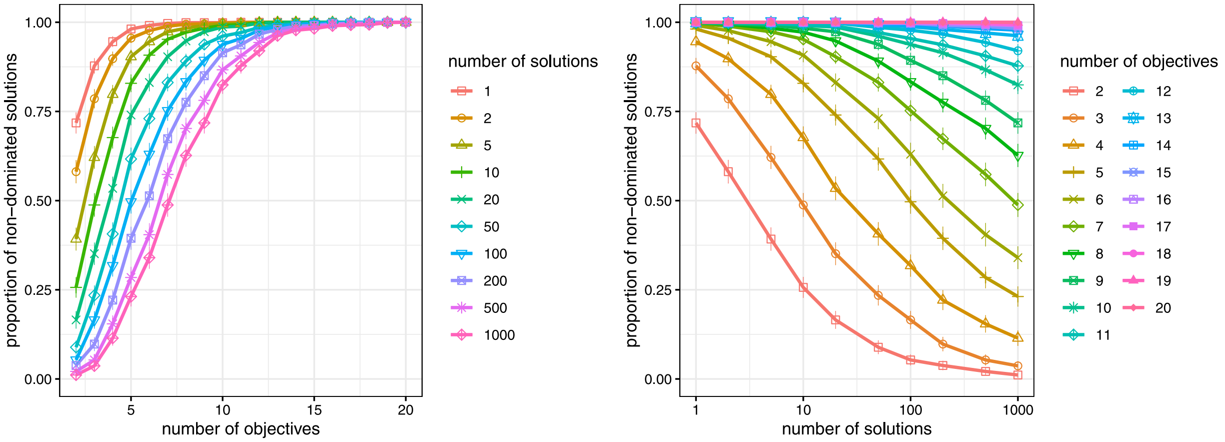

Fig. 4. Proportion of solutions that are not dominated by any of μ random solutions for multi-objective NK-landscapes, with respect to the number of objectives m (left), and to the number of solutions μ (right).

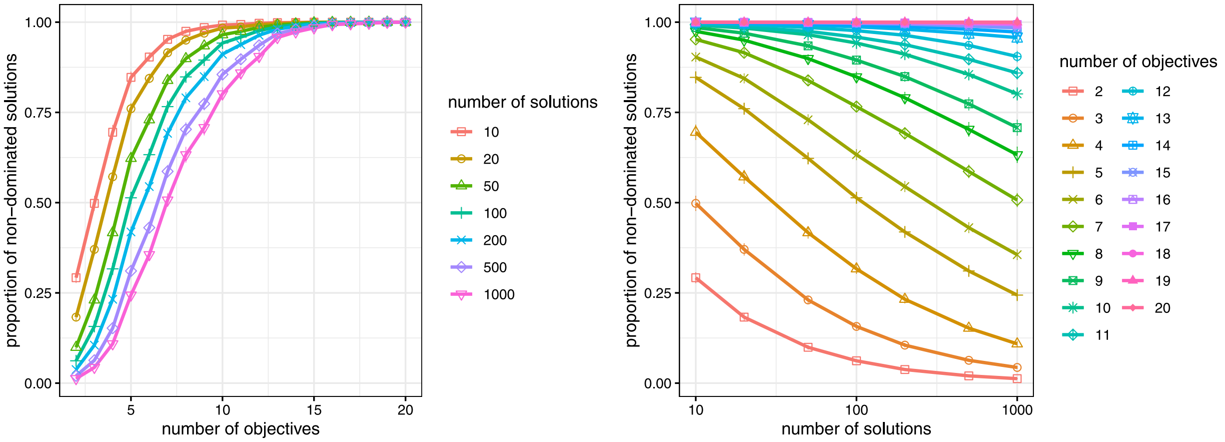

Fig. 5. Proportion of non-dominated solutions in random populations for multi-objective NK-landscapes, with respect to the number of objectives m (left), and to the number of solutions μ (right).

drug design (Small et al., 2011) and engineering (Terzijska et al., 2014). In the former, the objectives related to potency and side effects of drugs need to be validated experimentally, while the cost of manufacturing a drug or number of drugs in a drug combinations can be obtained quickly.

The first work we are aware of on the topic of heterogeneous objectives was by Allmendinger and Knowles (2013). This work was motivated by a real-world scenario where the objective function components were of different “latency” (evaluation times). This initial study was then extended in (Allmendinger et al., 2015), where different strategies for coping with latencies in a bi-objective problem were proposes and analyzed. Recently, more research has been carried out on the topic of heterogeneous objectives including the application of surrogate-assisted methods (Chugh et al., 2018; Wang

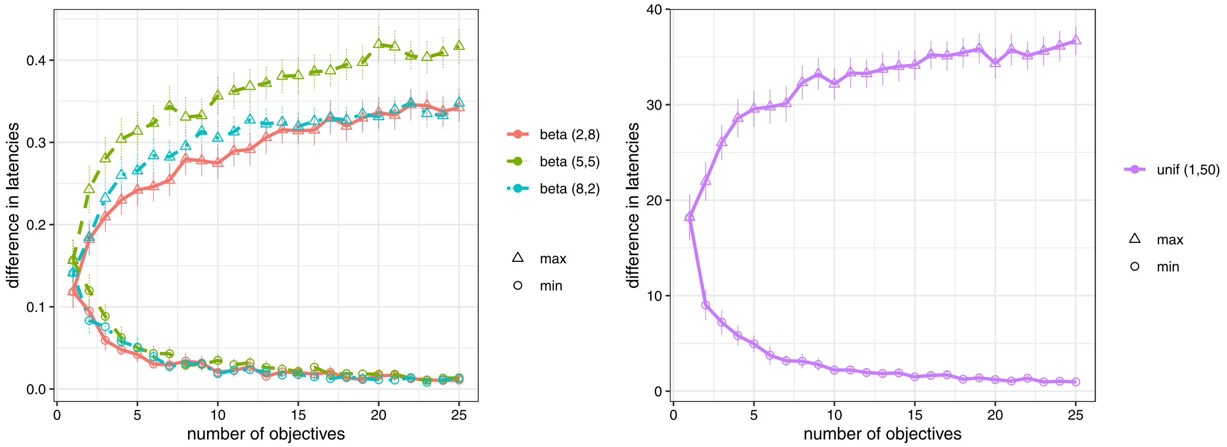

Fig. 6. Mean and standard error of minimum (circles) and maximum differences (triangle) in objective evaluation duration (y-axis) as a function of the number of objectives, m (x-axis). The left plot shows this data for three Beta distributions, and the right plot for a uniform distribution.

et al., 2020, 2021), transfer learning (Wang et al., 2020, 2021) and non-evolutionary methods (Thomann, 2019). The reader is referred to Allmendinger and Knowles (2021) for a more in depth review on heterogeneous objectives.

To motivate and further raise awareness of heterogeneous objectives, we investigate the relationship between the number of objectives in a problem and the degree of heterogeneity induced by them. For the purpose of this experiment, we limit ourselves to heterogeneity in form of latencies (evaluation times) across the objectives. Assuming a problem with a certain number of objectives, each being associated with an evaluation duration drawn from a given distribution, we want to know what is the mean minimum and maximum difference in evaluation times among these objectives. We repeated this experiment for different numbers of objectives (1–25 objectives as indicated on the x-axis) and four distributions to drawn the evaluation durations from. Each configuration was repeated 100 times, and the mean and standard error of the minimum and maximum differences in evaluation durations are plotted in Fig. 6. As the considered distributions, we considered (i) three different Beta distributions — $\mathtt{beta}(2,8)$ (skewed to the right), $\mathtt{beta}(8,2)$ (skewed to the left), and $\mathtt{beta}(5,5)$ (symmetric), each defined on the interval [0,1] — allowing us to simulate conveniently skewness in the evaluation of durations of the objectives, and (ii) one uniform distribution defined on the interval [1,50]. The Beta distribution has also been used extensively in the literature to quantify the duration of tasks (see seminal paper of Malcolm et al., 1959) in different contexts. The purpose of using the uniform distribution is to have a baseline to compare against.

It can be seen from the left plot of Fig. 6 that, regardless of the skewness and probability distribution, the mean minimum and maximum difference in evaluation durations of objectives starts roughly from the same level for a bi-objective problem ($m = 2$) with the two mean differences becoming exponentially more and more distinct as the problem becomes many-objective. This pattern is expected since it becomes more likely that the evaluation duration of a new objective (as sampled from the Beta distribution) is either more similar or distinct to the evaluation duration of an existing objective. Perhaps more surprisingly is the asymmetry between the minimum and maximum difference with increasing m: While the mean minimum difference is hardly affected by the choice of probability distribution (with the difference flattening quickly from around $m > 15$ objectives), there is a statistical difference between the Beta distributions when considering the mean maximum difference; in particular, the mean maximum difference associated with the symmetric distribution ($\mathtt{beta}(5,5)$) increases faster than with the two skewed distributions. This pattern stems from the fact that its more likely to sample extreme values for the evaluation duration with the symmetric distribution. Consequently, the gap between the mean minimum and maximum difference in evaluation times is even greater in the case of the uniform distribution (right plot of Fig. 6), where the probability of sampling any possible evaluation time is equal within the interval [1,50].

Having observed that the algorithm choice is affected amongst others by latency (Allmendinger and Knowles, 2021), this experiment indicates that knowing about the distributions of latency (evaluation times) can be used in the selection and design of novel algorithms to cope with heterogeneous objectives.

3.5. *Distance between solutions*

The distance between pairs of solutions randomly selected from the whole search space and from the Pareto set is reported in Fig. 7. We consider multi-objective NK-landscapes with $m \in \{2, 3, \dots, 20\}$. For each instance, we select 30 random pairs of solutions, and we report their Hamming distance in the design space and their Euclidean distance in the objective space. The Hamming distance among random solutions does not depend on the number of objectives, of course. However, the Hamming distance among Pareto optimal solutions increases significantly with the number of objectives, m. For $m > 13$ objectives, the expected distance among Pareto optimal solutions matches the one from random solutions. This comes as no surprise since most solutions are Pareto optimal in such high-dimensional objective spaces, as already mentioned in Section 3.1. However, when few objectives are considered, Pareto optimal solutions are much closer than random solutions. As such, for many-objective problems, good-quality solutions spreading along the Pareto front are far away from each other, so that we argue that few building blocks (if any) might actually be exploited by (blind) recombination.

When looking at the objective space, the Euclidean distance among objective vectors is increasing almost linearly with the number of objectives, with the distance among Pareto optimal solutions being only slightly lower than the distance among random solutions for $m < 15$ objectives. As can be visualized in the illustrative two-objective NK-landscapes provided in Fig. 8, reproduced from Verel et al. (2013), the objective space resembles a multidimensional “ball”, so that non-dominated objective vectors are not particularly closer than random ones. Given that the distance between solutions in the objective space increases with the number of objectives, we expect that identifying a high-quality representation of the Pareto front, in terms of coverage, gets more difficult for many-objective optimization problems.

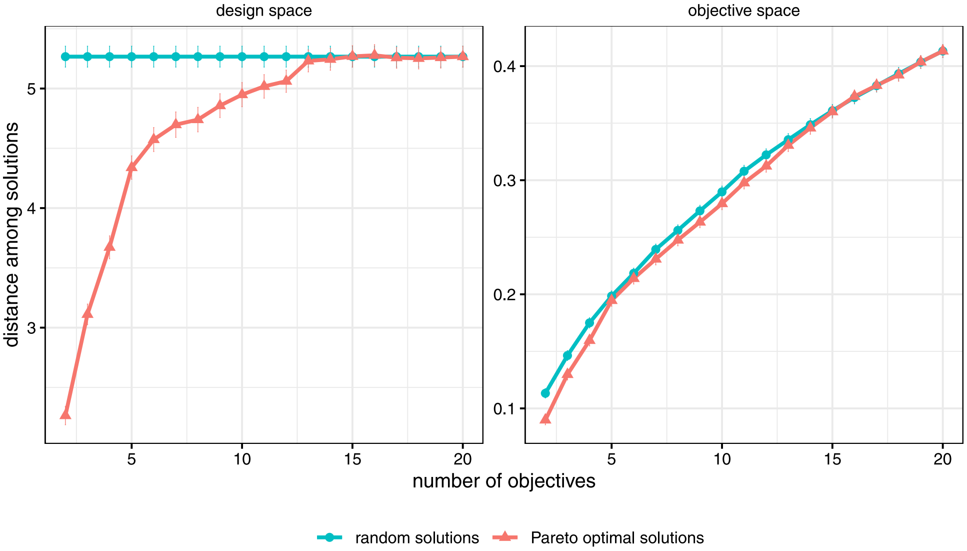

Fig. 7. Distance among solutions from the whole search space and from the Pareto set for multi-objective NK-landscapes with respect to the number of objectives.

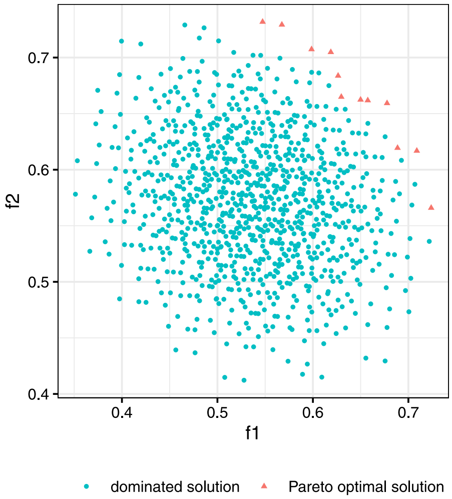

Fig. 8. Objective space of a multi-objective NK-landscape instance with $m = 2$ objectives.

4. Effect of the number of objectives on the complexity of multi-objective procedures and algorithms

In this section we report a number of computational experiments, both performed for the purpose of this paper and analyzing results reported in the literature. Note that, although a number of computational studies with the methods considered in this paper have been reported in the literature, they are usually focused on performance comparison, in most cases w.r.t. a newly-proposed method. By contrast, our goal is to more systematically study the influence of the number of objectives on the behavior of some representative algorithms. Furthermore, usually smaller numbers of objectives are used in reported experiments.

Although most studies are based on a number of pairwise comparisons of solutions, it is important to notice that the elementary operation for complexity results reported below is a pairwise comparison per objective. This choice is motivated by the fact that we want to highlight the effect of the number of objectives (m) on different multi-objective optimization tools and methods.

4.1. Dominance test and updating the Pareto archive

In this section, we consider the processes of testing if a solution $\mathbf{x}$ is dominated or not by a Pareto archive and of updating this archive. Updating the Pareto archive A with a new solution $\mathbf{x}$ means that all solutions dominated by $\mathbf{x}$ are removed from A and $\mathbf{x}$ is added to A if it is not dominated by any solution in A. The complexity analysis of the two processes is the same, since the dominance test is the bottleneck part of the updating process.

The simplest data structure for the Pareto archive is a simple, unordered list of solutions with linear time complexity of update. Several methods and related data structures aiming at efficient realization of the Pareto archive update have been proposed, e.g., Quad Tree (Sun and Steuer, 1996; Mostaghim and Teich, 2005; Sun, 2006, 2011; Fieldsend, 2020), MFront II (Drozdík et al., 2015), BSP Tree (Glasmachers, 2017), and ND-Tree (Jaszkiewicz and Lust, 2018). Jaszkiewicz and Lust (2018) reported some complexity results for ND-Tree:

- Worst case: $\mathcal{O}(m\ N)$;
- Best case: $\Theta(m\ \log(N))$;
- Average case: $\Theta(m\ N^b)$, where N is the size of the archive, and $b \in [0, 1]$ is the probability of branching (i.e. the probability of testing more than one subnode in ND-Tree).

Note that linear time complexity in the worst case holds also for Quad Tree (Sun and Steuer, 1996) and the sublinear time complexity in average case could also be obtained with BSP Tree (Glasmachers, 2017).

The Pareto archive may be either bounded in size or unbounded, i.e. contain only some or all non-dominated solutions generated so far (Fieldsend et al., 2003). In the latter case, the size of the Pareto archive may, in general, grow exponentially with the number of objectives (see Section 3.1). Assuming that $N = \mathcal{O}(c^{m-1})$ (see Section 3.1), the complexity of the update process of an unbounded archive becomes:

- Worst case: $\mathcal{O}(m\ N) = \mathcal{O}(m\ c^{m-1})$;
- Best case: $\Theta(m\ \log(N)) = \Theta(m\ \log(c^{m-1})) = \Theta(m^2\ \log(c))$;
- Average case: $\Theta(m\ N^b) = \Theta(m\ c^{(m-1)b})$.

In other words, in the worst case, the time grows exponentially with m, and, in the average case, the time may grow exponentially with m if $c^b > 1$.

To illustrate the above analysis, we perform the following computational experiment. We consider multi-objective NK-landscapes with $n = 16$, $k = 0$, and $m \in \{3, 4, \dots, 20\}$. For each setting, we generate 30 random instances. All $2^{16} = 65\,536$ solutions are processed in random order. Note that in this experiment we use instances with $n = 16$ instead of $n = 10$ used in other places, because with $n = 10$ the number of solutions was too low to show significant differences between the evaluated methods. The methods used in this experiments are simple list, ND-Tree (Jaszkiewicz and Lust, 2018), Quad Tree (precisely Quad Tree 2 algorithm described by Mostaghim and Teich, 2005 with the corrections described by Fieldsend, 2020), and MFront II (Drozdík et al., 2015) with the modifications proposed by Jaszkiewicz and Lust (2018). We used C++ implementations of these methods described by Jaszkiewicz and Lust (2018), however, implementation of the Quad Tree has been improved on both technical level and using the corrections proposed by Fieldsend (2020). Fig. 9 (left) presents average running times of the four methods needed to process all solutions. As could be observed, the ranking of the methods depends on the number of objectives in a non-trivial way. Overall, the two best methods are Quad Tree and ND-Tree, with Quad Tree being better for $m = \{12, \dots, 17\}$ objectives, and ND-Tree being better in other cases. Somehow surprisingly, simple list performs very well for small numbers of objectives and is the best method for $m = 3$ and $m = 4$ objectives. This is related to the design of the experiment in which all solutions are used. For small numbers of objectives, there are relatively few non-dominated solutions (see Fig. 1), so the list is relatively short and a new solution has a high chance to be dominated by many solutions in the list. Thus, a dominating solution is often quickly found and the update process is finished. Note also, that MFront II is the worst performing method.

It could also be observed that the running time grows exponentially with the number of objectives. This could be however, caused just by the growing size of the Pareto archive (see Section 3.1, Fig. 1). Thus, in Fig. 9 (right) we present running times divided by the number of Pareto optimal solutions. The relative running time changes only slightly for Quad Tree and ND-Tree again with a non-trivial pattern.

Furthermore, as we discussed above in this section, methods like ND-Tree aim at ensuring sublinear time complexity w.r.t. the size of the Pareto archive. To verify this, we measure running times needed to achieve the Pareto archive of $10\%, 20\%, \dots, 100\%$ of the final size. To be precise, the running time reported e.g. for 20% is the running time from the moment of achieving 10% of the final size till the moment of achieving 20% of the final size divided by the number of processed solutions (in other words, we report average time of processing a single solution in a given interval of the Pareto archive size). Ideally, this running time should be constant or grow sublinearly with the size of the Pareto archive. The results for 20 objectives are presented in Fig. 10. Since the running times of the four methods differ significantly (see Fig. 9), the running times of each method are normalized such that the maximum time is 1. As it can be observed, running time needed to process a single solutions for 20 objectives grows linearly with the size of the Pareto archive for all methods. Of course, the speed of this growth differs for different methods. On the other hand, for some smaller numbers of objectives the running times are indeed almost constant or clearly sublinear (see Fig. 11[2]). This means that for each method, for some number of objectives, we observe a switch of its behavior from sublinear to approximately linear dependence on the size of the Pareto archive. The switching point is different for different methods and is between 5–7 objectives for the simple list and MFront II, between 6–8 objectives for ND-tree, and between 4–6 objectives for Quad tree. In other words, since the number of Pareto optimal solutions grows in general exponentially with the number of objectives (see Section 3.1), according to the presented experiment, for at least 8 objectives we can expect exponential growth (w.r.t. m) of updating an unbounded Pareto archive independent on the method used. In our opinion, it is caused by the reduced discriminative power of the dominance relation (see Section 3.2) for higher numbers of objectives. Although different methods are based on different ideas, all of them use some properties of dominance relation to speed-up the update process, thus all of them are affected by its reduced discriminative power.

The methods tested in our experiment together with several other methods were also recently experimentally evaluated by Fieldsend (2020). The conclusion of that experiment was that ND-Tree generally performs best in long runs, but this was not a universal finding. Furthermore, the performance of the different methods can vary considerably between hardware architectures, and in a non-constant fashion. Also, our additional preliminary experiments (not reported in this paper) indicate that performance may strongly depend on the data sets used and the order in which the solutions are processed. In particular, the running time may depend on whether the new solution is dominated, non-dominated or dominating, and the pattern may be different for different methods. Thus, we encourage further experiments testing practical behavior of the methods for updating the Pareto archive, in particular using different orders of solutions, which is however, out of the scope of this paper.

4.2. Computing and approximating hypervolume

When assessing the performance of multi-objective optimization algorithms, or in indicator-based evolutionary multi-objective optimization, the indicator-value of a set of solutions is to be computed multiple times. One of the recommended and most-often used indicator is the hypervolume (Zitzler et al., 2003; Shang et al., 2021), because of its compliance with the comparison of sets of points based on the dominance relation. Note that, since dominated solutions do not influence the hypervolume, only non-dominated solutions, forming a Pareto archive of size N, need to be taken into account. Unfortunately, the exact hypervolume computation time is known to grow exponentially with the number of objectives. Although efficient algorithms exist for $m = 3$ (Beume et al., 2009) and $m = 4$ (Guerreiro and Fonseca, 2018), the best known algorithm for the general case has a time complexity of $\mathcal{O}(N^{m/3}\ \text{polylog}\ N)$ (Chan, 2013).

To analyze the above theoretical results experimentally, we test the behavior of two state-of-the-art methods for exact computation of hypervolume, namely the non-incremental version of the Hypervolume Box Decomposition Algorithm (HBDA-NI) (Lacour et al., 2017) and the Improved Quick Hypervolume algorithm (QHV-II) (Jaszkiewicz, 2018). The results published in these papers are used, and the running times of HBDA-NI have been divided by 2.5 to compensate processor differences as suggested in Jaszkiewicz (2018). The results are reported for data

[2] The ‘jump’ or running time observed for Quad tree with 20% of the Pareto archive size and small number of objectives ($m = 3$ in particular) is probably because the initial tree must be built before the method achieves full performance.

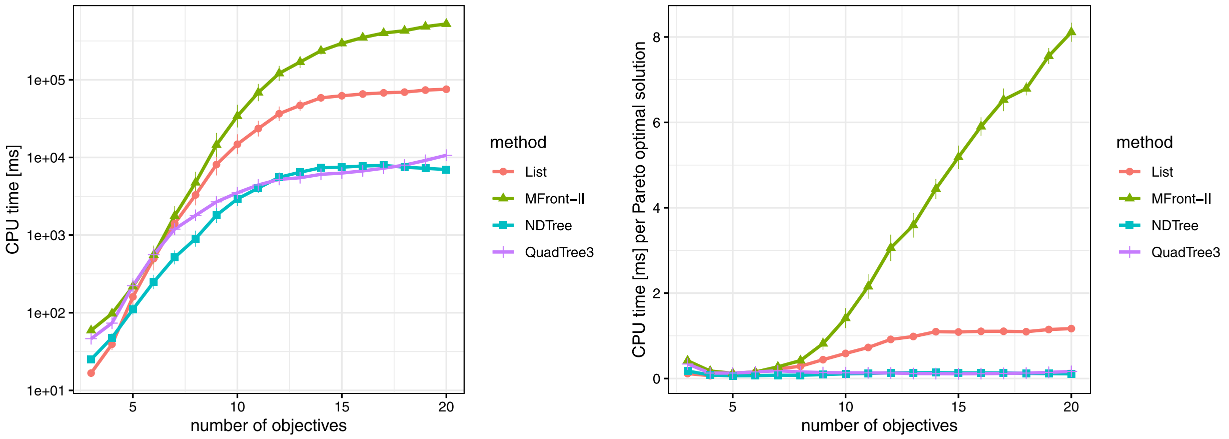

Fig. 9. Running time of four methods for updating the Pareto archive (left, logarithmic scale), and running time divided by the number of Pareto optimal solutions (right).

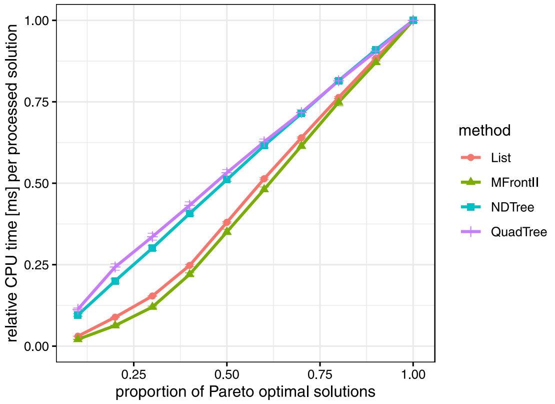

Fig. 10. Relative running time of Pareto archive for processing a single solution w.r.t. to the Pareto archive size for $m = 20$ objectives.

sets proposed by Lacour et al. (2017) (concave, convex, and linear) composed of 1 000 points in $m \in \{4, 6, 8, 10\}$ dimensional objective spaces, with objective values (and thus hypervolume) normalized to the range $[0, 1]$. We use 10 instances for each combination of data set type and number of objectives. The results are presented in Fig. 12. Note that for the running time w.r.t. the number of objectives, a logarithmic scale is used. The running times of the exact methods grow indeed exponentially with the number of objectives, which suggests that the exponential time complexity holds not only in the worst case, but in the typical case as well. At the same time the running times grow relatively slowly with the number of points (see Fig. 12, right).

Alternatively, the hypervolume can be approximated by Monte Carlo sampling (Bader and Zitzler, 2011). In this case, the complexity is $\Theta(s\ m\ N^b)$, where s is the number of sampling points and the term N^b relates to the dominance tests (see Section 4.1). Since Monte Carlo sampling is just a sequence of s independent experiments, each asking a yes/no question (dominated/non-dominated), the confidence interval can be derived from the binomial distribution and does not depend on the number of objectives. On the other hand, the question remains if the size of the confidence intervals should be reduced with growing number of objectives m and/or growing N.

To analyze the above question, we consider first the case where the size N of the set of points for which hypervolume is estimated is constant and only the number of objectives changes. The required precision of hypervolume estimation could be related to an average contribution of a single point to the hypervolume, i.e. the difference in the hypervolume with and without this single point. In other words, intuitively speaking, the precision of hypervolume estimation should be sufficient to distinguish addition/subtraction of a given number of points. It could be expected that the average contribution does not depend on the number of objectives. To analyze this experimentally, we use the same data sets as above. For each individual point its contribution to the hypervolume of the given set is calculated with an exact method. The results presented in Fig. 13 are average contributions for $10 \times 1\,000 = 10\,000$ individual points. The influence of the number of objectives depends mainly on the instance type and only in the case of convex instances the average contribution decreases with a growing number of objectives. However, note that for convex instances, the hypervolume approaches 0 as the number of objectives grows (i.e. the points are very close to the nadir point), thus the contribution of an individual point also approaches 0. On the other hand, the average contribution grows with the number of objectives for linear instances. Summarizing, no general trend for the decrease or increase of the average contribution with the growing number of objectives could be concluded from this experiment, thus we conclude that also the size of the confidence interval should not change with the number of objectives and constant N.

Additionally, as discussed above, it is often assumed that N increases with m. In this case, it could be expected that the average contribution is $\Theta(1/N)$. To test this assumption, we show the results in Fig. 13 for the same set of instances with $200, 400, \dots, 1\,000$ points. As it can be observed, the average contribution indeed decreases approximately linearly with $1/N$. Since we propose to treat the average contribution as the indicator of required precision, the size of the confidence interval r (i.e. the difference between the upper and lower confidence bound) for the estimated hypervolume should be $\Theta(1/N)$. The confidence interval for binomial distribution could be calculated with Wilson score interval with continuity correction for binomial distribution (Newcombe, 1998). According to this test, $r = \Theta(1/\sqrt{s})$, where s is the number of sampled points, so $s = \Theta(1/r^2)$. Thus, in order to achieve the required accuracy, the number of sampled points s should be $\Theta(N^2)$.

To test the above hypothesis, we perform the following experiment. Using the same data sets we run Monte Carlo sampling until it achieves size of the confidence interval lower or equal to $5/N$. We use the simplest version of Monte Carlo sampling in which the hypercuboid defined by the nadir and ideal points is uniformly sampled. ND-tree is used for the dominance test. The results are presented in Fig. 14. As can be observed, for linear and concave instances the running times grow slowly with the number of objectives, which is due to the increasing number of comparisons of objective values. For convex instances the running time of Monte Carlo sampling decreases with the growing number of objectives, because for this set of instances with high numbers of objectives, the hypervolume values are very close to 0 which makes the estimation task easier. We test also the influence of the

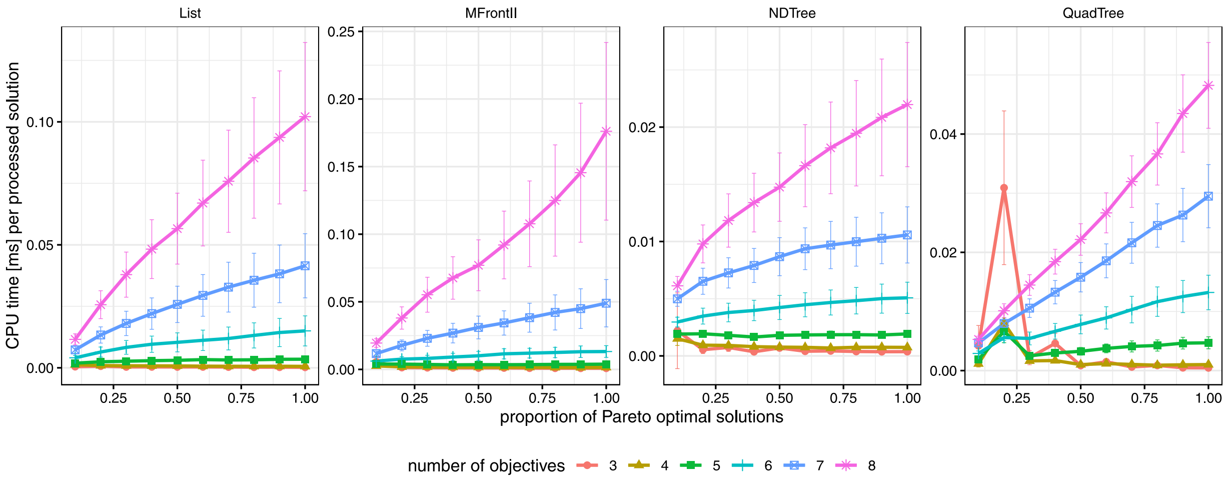

Fig. 11. Running time of Pareto archive for processing a single solution w.r.t. to the relative Pareto archive size for smaller numbers of objectives $m \in \{3, 4, \dots, 8\}$.

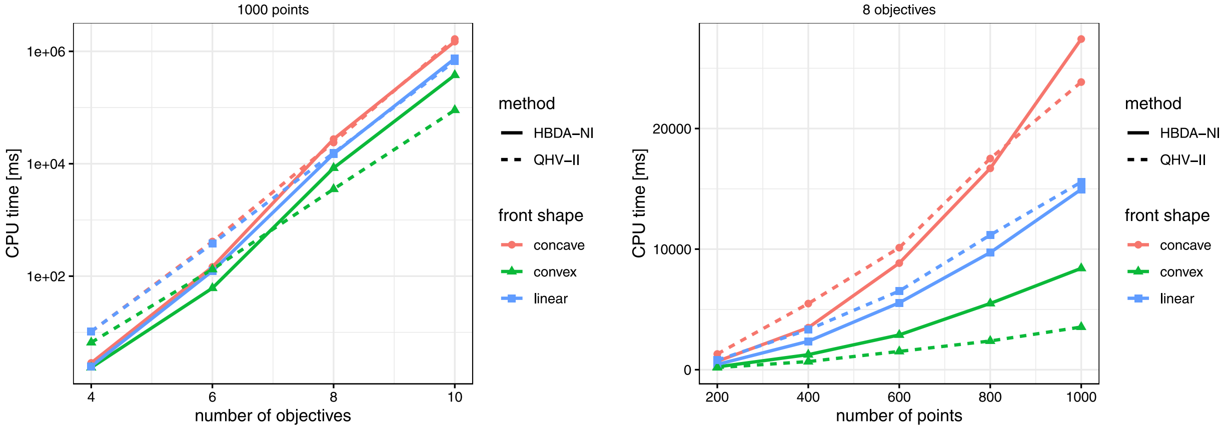

Fig. 12. Running time of exact hypervolume computation methods w.r.t. the number of objectives (left, logarithmic scale) and the number of points for instances with $m = 8$ objectives (right).

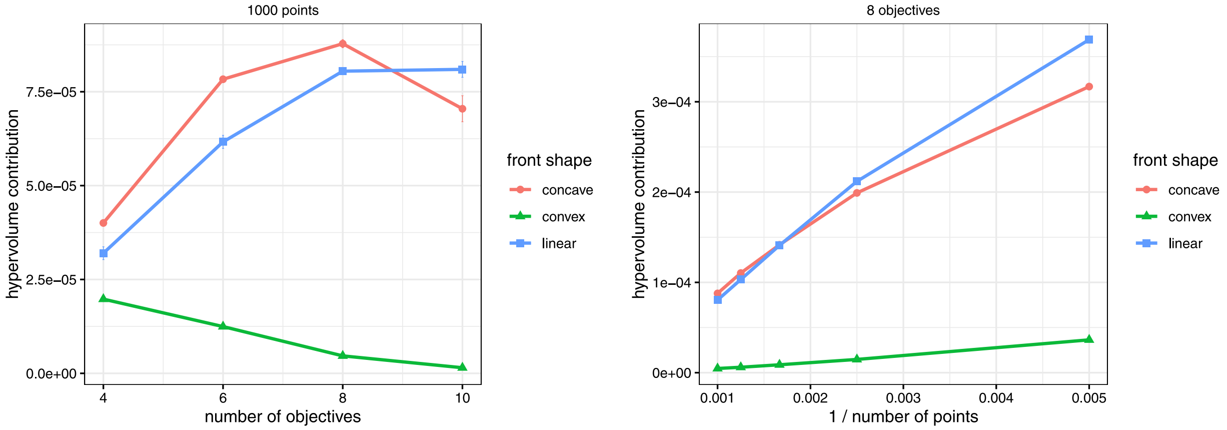

Fig. 13. Average contribution of individual solutions to the hypervolume w.r.t. the number of objectives (left) and the number of points (right).

number of points on the running time for 8 objective instances (Fig. 14) together with quadratic trend lines which fit the data very well. Note, however, that the running times of exact methods are shorter than that of Monte Carlo sampling for 4 objectives and comparable for 6 objectives. Only for 8 and 10 objectives Monte Carlo sampling is clearly beneficial in the presented experiment. Furthermore, since the required sample size, and thus CPU time, is $\Theta(1/r^2)$, the running time of Monte Carlo may be much higher if a smaller confidence interval is required. For example, if the required confidence interval is $1/N$ (i.e. 5 times smaller), the running time of Monte Carlo sampling would grow about 25 times and the running times of exact methods would be larger only for 10 objectives.

Summarizing, both the theoretical analysis and the presented experiments suggest that Monte Carlo sampling is an interesting alternative to exact methods for many-objective problems, if an approximate value of hypervolume is sufficient in a given context. Note also that recently

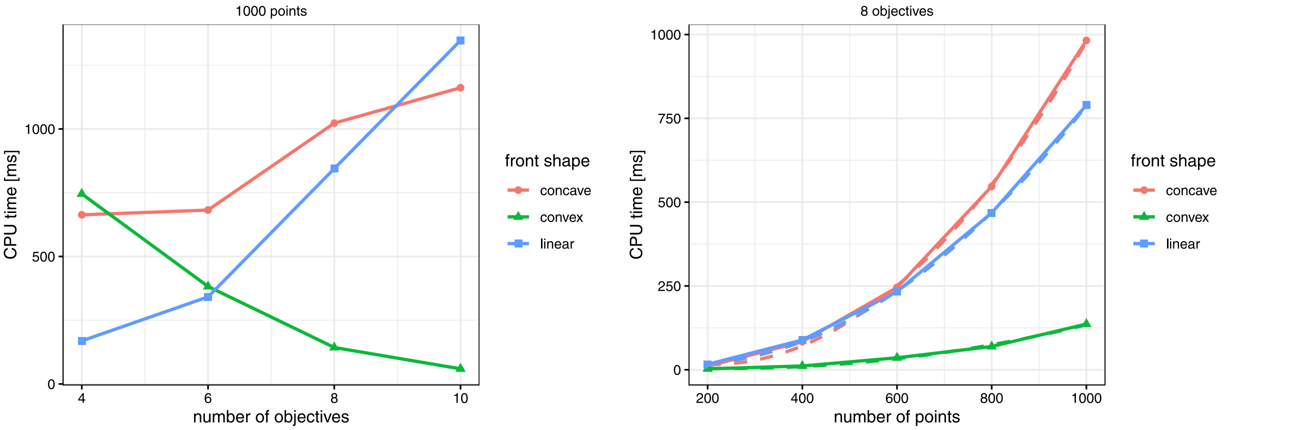

Fig. 14. Running time of Monte Carlo sampling for hypervolume estimation w.r.t. the number of objectives (left) and the number of points (right).

a number of more advanced algorithms for approximate hypervolume computation have been proposed, some of them providing an approximation guarantee (Ishibuchi et al., 2009; Tang et al., 2017; Ma et al., 2018; Shang et al., 2018; Jaszkiewicz et al., 2020; Zhang and Golovin, 2020).

4.3. Impact on scalarization methods

The aim of this section (see also the Appendix) is to discuss the influence of the number of objectives in the scope of scalarization procedures, which are a well-established approach to MO optimization; hence, we can afford being more technical in our discussion. Using the properties of a general scalarization function, we derive corresponding properties of some prominent scalarization techniques as special cases of the general scalarization function.

We consider a MO problem with the vector-valued objective function $\mathbf{f}$ and the feasible set X (see Definition 2.1). An important question is how to characterize the sets of Pareto (weakly) optimal elements of MO problems via suitable scalarization methods. In the non-convex case, especially in combinatorial MO, it is important to provide suitable non-linear scalarization functions. Depending on the monotonicity properties of the involved scalarization functions, some of the scalarization methods used for deducing characterizations of solutions to MO problems generate weakly Pareto optimal solutions (see Definition 2.3), other scalarization methods generate Pareto optimal solutions (see Definition 2.2). For an overview on scalarization methods and corresponding algorithms for solving MO problems, we refer to Eichfelder (2008), Tammer and Weidner (2020) and references therein.

Pascoletti and Serafini (1984) introduced a scalar optimization problem that can be formulated using a general non-linear scalarizing function that we will discuss in this section. Corresponding non-linear scalarization methods, including applications in economics, are studied by Bonnisseau and Cornet (1988), Bonnisseau and Crettez (2007) and Luenberger (1992a,b). The parameters in the scalar optimization problem by Pascoletti and Serafini (1984) are elements $\ell \in \mathbb{R}^m$ and $a \in \mathbb{R}^m$, see Eq. (A.11). Furthermore, in the ϵ-constraint problem by Haimes et al. (1971) and Chankong and Haimes (1983), parameters $\epsilon = (\epsilon_1, \ldots, \epsilon_m) \in \mathbb{R}^m$ are involved. The scalarization method by Pascoletti and Serafini (1984) and the ϵ-constraint problem are used for deriving parameter-based adaptive algorithms in Eichfelder (2008), compare also Polak (1976). By varying the parameters $\ell \in \mathbb{R}^m$ and $a \in \mathbb{R}^m$ in Eq. (A.11), adaptive algorithms generate an equidistant approximation of the set of Pareto optimal elements to the MO problem by solving scalarized problems. If the number of objectives m is increasing, then the parameter vectors ℓ and a belong to higher dimensional spaces, i.e., the number of preference parameters increases linearly.

4.3.1. A general scalarization method by translation invariant functions

Let us consider the general class of scalarization functionals given by translation invariant functions $\varphi : \mathbb{R}^m \to \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$. The translation invariance of φ along a direction $\ell \in \mathbb{R}^m \setminus \{0\}$ is defined by the property $\varphi(y + t\ell) = \varphi(y) + t$ for all $y \in \mathbb{R}^m$, $t \in \mathbb{R}$. Translation invariant functions play an important role in scalarization methods for multi-objective optimization problems as well as in risk theory and mathematical finance (Artzner et al., 1999; Jaschke and Küchler, 2001). Translation invariant functions can be formulated using a certain set $A \subset \mathbb{R}^m$ and a parameter vector $\ell \in \mathbb{R}^m \setminus \{0\}$ with the property $\ell \in -0^+A \setminus \{0\}$ ($0^+A := \{y \in \mathbb{R}^m \mid A + \mathbb{R}_+ y \subseteq A\}$ denotes the recession cone of A). An equivalent formulation of translation invariant functions using A and ℓ is given in Eq. (A.3).

We explain that the scalarization functions involved in the Chebyshev scalarization, the ϵ-constraint problem and other scalarization methods are special cases of the general translation invariant scalarization function (A.3). Using these results, it is possible to obtain continuity, monotonicity, convexity and other algebraic properties of the scalarization functions in the Chebyshev scalarization, the ϵ-constraint problem from the corresponding properties of the general nonlinear translation invariant function derived in Göpfert et al. (2003, Theorem 2.3.1). These properties are very important for a characterization of the solutions to MO problems.

In Corollary A.1, especially in Eq. (A.10), we show that the (along ℓ translation invariant) function $\varphi_{A,\ell} := \varphi$ is completely defined by the parameter vectors $\ell \in \mathbb{R}^m \setminus \{0\}$ and $w \in \mathbb{R}^m$ and elements $a^i \in \mathbb{R}^m$ $(i = 1, \ldots, n)$. These parameter vectors are involved in certain inner products of the m dimensional vectors ℓ and w included in the description of the function $\varphi_{A,\ell}$ in Eq. (A.10). If we add some more objectives, then the scalarized problem is not more difficult because the number of objectives m is only involved in the inner products $\langle a^i, w\rangle$ and $\langle a^i, \ell\rangle$ of the m dimensional vectors $a^i, \ell, w \in \mathbb{R}^m$ taking into account Corollary A.1.

4.3.2. Scalarization methods for multi-objective combinatorial optimization

The application of scalarization methods, especially of the ϵ-constraint method, for combinatorial MO problems is discussed by Figueira et al. (2017). As explained in (a) of the Appendix, the ϵ-constraint method is a special case of a scalarization by means of translation invariant functions $\varphi_{A,\ell}$ (see Eq. (A.3)) with the parameters A and ℓ given by Eqs. (A.4)–(A.5). Scalarizations of combinatorial MO problems with an objective function $\varphi_{A,\ell} \circ \mathbf{f}$, where A and ℓ are given by Eqs. (A.4)–(A.5), are resource-constrained combinatorial problems. These scalar combinatorial optimization problems constitute to be NP-hard. In particular, if we consider combinatorial problems, then the additional ϵ-constraints can make the optimization problem with the objective function $\varphi_{A,\ell} \circ \mathbf{f}$ based on different parameters A and ℓ like in Eqs. (A.4)–(A.5) considerably more difficult to solve than the corresponding weighted sum scalarization of the MO problem. Figueira

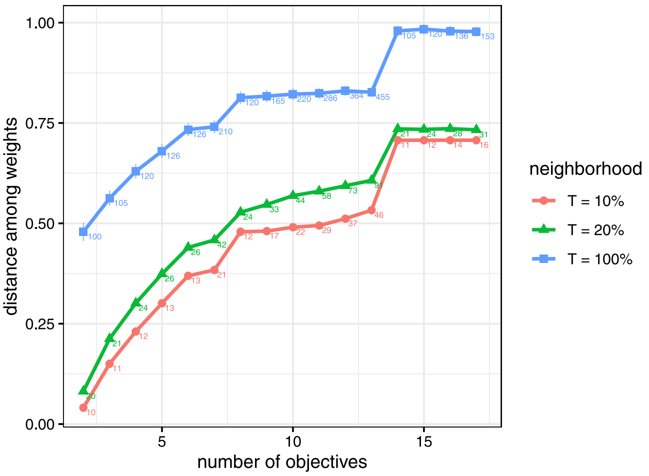

Fig. 15. Distance among weight vectors with respect to the number of objectives. Below each point is reported the number of weight vectors for the considered setting.

et al. (2017) explained that the cause for this supplementary difficulty is the perturbation of the combinatorial structure of the polyhedron of feasible elements by the ϵ-constraints (see Eqs. (A.2)–(A.4)).

Furthermore, it is mentioned in Figueira et al. (2017) that there are particular cases where the property of total unimodularity is compatible with the ϵ-constraint method. There are combinatorial MO problems, where total unimodularity can be maintained during the scalarization procedure. For instance, the constraint matrix of a binary knapsack problem or the binary assignment problem is totally unimodular.

In general, it seems to be preferable to use the non-linear translation invariant function in Eq. (A.10) for scalarization. Taking into account Corollary A.1, it is possible to avoid additional ϵ-constraints.

Concerning the influence of the number of objectives:

- **ϵ-constraint method:** There are more ϵ-constraints in the scalarized problem in the case that we add some objectives; see Eqs. (A.2)–(A.4). Furthermore, the structure of the MO problem's constraints can be perturbed by the additional ϵ-constraints.
- **scalarization by $\varphi_{w+A_L,\ell}$:** If we add some more objectives, then the problem (MO($\varphi_{w+A_L,\ell}$)) in the Appendix is not more difficult because the number of objectives m is only involved in the inner products $\langle a^i, w\rangle$ and $\langle a^i, \ell\rangle$ for $a^i, w, \ell \in \mathbb{R}^m$; see Eq. (A.10) in Corollary A.1.

4.4. Distances between weight vectors

In decomposition-based MOEAs, such as MOEA/D (Zhang and Li, 2007) and its variants (Trivedi et al., 2017), multiple scalar sub-problems are optimized simultaneously. Each sub-problem is defined by a particular weight vector setting for the considered scalarizing function. It is often argued that the search process actually benefits from solving the sub-problems cooperatively, given that there exists a certain locality among them, the solution from one sub-problem contributing to the solution of another sub-problem and vice-versa. In particular, a neighborhood is defined among sub-problems in order to limit the exchange of information between them. In this section, we investigate how the distance between uniformly-defined weight vectors is impacted by the number of objectives.

We generate weight vectors following a simplex-lattice design (Scheffé, 1958) as performed, e.g., in the original MOEA/D (Zhang and Li, 2007), where the number of weights is set as the population size μ. The simplex-lattice design generates μ weight vectors such that $\mu = \binom{H+m-1}{m-1}$, where H is a user-defined parameter. In our experiments, for a given number of objectives m, we take the smallest $H-$ value such that $\mu \geq 100$. The neighborhood of a given weight vector is defined as the set of the T closest weight vectors, based on Euclidean distance. We consider three neighborhood sizes $T \in \{10\%, 20\%, 100\%\}$, given as a proportion of the total number of weight vectors μ. When $T = 100\%$, this means that there is no restriction on the exchange of information between solutions; i.e. the neighborhood is made of the entire population. For a given setting, we select 900 pairs of weight vectors randomly (we used the same setting in Section 3.5), each time within a given neighborhood, and we report the average Euclidean distance between them and confidence intervals in Fig. 15. Note that the 'jump' between $m = 13$ and $m = 14$ results from the fact that, with the simplex-lattice design, only some sparse values of μ may be obtained, and we select the smallest $H-$ value such that $\mu \geq 100$. The value of μ obtained in this way is much smaller for $m = 14$ (105) than for $m = 13$ (455) and the distances between the weight vectors become larger also due to this fact.

We observe that the distance among weight vectors increases with the number of objectives. When no neighborhood is considered ($T = 100\%$), the distance goes from about 0.5 for $m = 2$ to close to 1 for $m \geq 14$. Restricting the selection of weight vectors among the 10% or 20% closest ones reduces the distance to an order of magnitude. However, the trend remains similar, and the distance exceeds 0.5 for problems with more than 12 objectives, even with a small neighborhood size of $T = 10\%$. As such, the assumption that the neighboring sub-problems share similar information becomes less accurate as the number of objective grows, and might actually affect the performance of decomposition-based MOEAs for many-objective problems.

5. Conclusions and future work

5.1. Summary

This paper has carried out a theoretical and empirical analysis of the impact of increasing the number of objectives m on (i) the characteristics of a multi-objective (MO) problem and (ii) the complexity of commonly used MO procedures and algorithms. For the empirical analysis, we used multi-objective NK-landscapes, which allowed us to conveniently scale up the number of objectives m. Table 1 provides an overview of the topics we have made contributions to, and where these can be found in the paper. The main findings of our analysis in terms of scaling efficiencies can be summarized as follows.

- **Good** scaling behavior (i.e., polynomial):
  - *Impact on scalarization methods:* We have proposed a general non-linear scalarizing function that can be used to describe several well known scalarizing techniques including weighted Chebyshev scalarization, weighted sum scalarization, Pascoletti–Serafini problem, ϵ-constraint problem,

and scalarization by Kaliszewski. Using the proposed nonlinear functions, the complexity of solving the corresponding scalar problem grows linearly with m.
  - *Approximating hypervolume:* We have shown that the complexity of computing the hypervolume exactly grows exponentially with m in the typical case (not only in the worst case). When approximating the hypervolume using Monte Carlo sampling, we demonstrated empirically that the size of the confidence interval (of the approximated hypervolume value) should not change with m assuming a constant Pareto archive size N; instead the number of sampled points should be $\Theta(N^2)$. Moreover, we showed that there is a switching point ($m \geq 8$ in our study) beyond which the running time of a sampling-based approach outperforms an exact method, and the switching point moves to higher m as the required confidence interval reduces in width.

- **Relatively good** scaling behavior:
  - *Dominance test and updating the Pareto archive:* Although, in the case of unbounded Pareto archive, the running time of updating the Pareto archive grows exponentially with m for all four considered data structures – a simple (unordered) list, ND-Tree, Quad Tree, and MFront II –, experiments reveal that ND-Tree and Quad Tree are more robust. With regards to the running time of processing single solutions w.r.t to the Pareto archive size, we observed empirically a switching point in terms of m beyond which the running time turns from sublinear to approximately linear dependence on the Pareto archive size. This switching point is between 5–7 objectives for the simple list and MFront II, between 6–8 objectives for ND-tree, and between 4–6 objectives for Quad tree.

- **Poor** scaling behavior (i.e., exponential complexity, decreased quality):
  - *Number of Pareto optimal solutions:* We have confirmed empirically that the number of Pareto optimal solutions grows exponentially with m, with less than 5% of solutions being Pareto optimal for $m = 2$ and this proportion growing to about 50% for $m = 7$, and more than 99% for $m = 20$ for the considered problems.
  - *Probability for a solution to be non-dominated:* We derived theoretically the probability that μ random pairs of solutions are mutually non-dominated, and this probability is $\left(1 - \frac{1}{2^{m-1}}\right)^{\mu}$. An empirical analysis confirmed the correctness of this probability, and showed that it becomes very likely that all solutions are mutually non-dominated for $m > 15$, even for values of $\mu > 1\,000$.
  - *Probability of having heterogeneous objectives:* We have shown experimentally that the level of heterogeneity among objectives in a many-objective problem increases with m. In particular, for the case of heterogeneous evaluation durations of objectives, with durations being drawn from a Beta distribution, we observed that the difference in the minimum and maximum difference in evaluation durations decreases and increases exponentially with m, respectively. A unskewed (symmetric) Beta distribution was associated with the largest increase in the maximum difference.
  - *Distance between solutions:* We have shown empirically that, as m increases, the expected distance (in the design space) among Pareto optimal solutions becomes more similar to the one among random solutions, and for $m \geq 15$ the distances were identical. Due to the spherical distribution of solutions in the objective space for the problems considered, the distance between Pareto optimal and random solutions in the objective space is very similar and increases linearly with m. These observations suggest that, for many-objective problems, (i) it becomes more difficult to discover a high-quality representation of the Pareto front as the Pareto optimal solutions are distant from each other, and (ii) that few building blocks (if any) might actually be exploited by (blind) recombination.
  - *Computing hypervolume exactly:* We have shown empirically that existing theoretical results hold and that the exact hypervolume computation time growths indeed exponentially with m in the typical case (not only in the worst case).
  - *Distance between weight vectors:* We observed empirically that the distance among uniformly-defined weight vectors increases with m. A smaller neighborhood size follows the same general trend but reduces the distance between weight vectors to an order of magnitude, with the distance exceeding 0.5 for $m > 12$ even for a neighborhood size of 10%. This implies that, as m increases, the assumption that neighboring sub-problems share similar information becomes less accurate.

5.2. Recommendations for MOEAs

This section will make use of our theoretical and empirical findings to provide recommendations and considerations on design choices for different multi-objective optimization paradigms when applying them to problems with many objectives.

- For **all classes of MOEAs**:
  - Algorithm performance should be evaluated in terms of representation quality of the Pareto front. This indicates an algorithm's ability to deal efficiently with the large distances between Pareto optimal solutions when m is large.
  - The distance between solutions in the objective space increases with m, causing the quality of representation to decrease and reduce coverage.
  - The level of heterogeneity in a MO problem increases with m, urging the need for customized methods for coping with heterogeneity for many-objective problems.
  - The distance between solutions in the design space increases with m, causing (blind) recombination to be less effective.
  - When no external Pareto archive is considered, the population size shall increase to maintain the same level of coverage of the Pareto front because the number of Pareto optimal solutions increases with m.
  - When an external Pareto archive is considered, a data structure such as Quad Tree or ND-Tree shall be used when m is large and in need to discover a good representation quickly.

- For **dominance-based MOEAs** (e.g. NSGA-II Deb et al., 2002):
  - The dominance relation becomes less discriminative as m increases, causing a reduction in the selection pressure (when selection is done based on dominance) and affect negatively the representation quality of the Pareto front.
  - The diversity maintenance method employed is critical to ensure adequate selection pressure as m increases, since most solutions in the search space are mutually non-dominated.

- For **decomposition/scalarization-based MOEAs** (e.g. MOEA/D Zhang and Li, 2007):
  - Assuming a constant number of weight vectors (and population size), all issues mentioned above for constant population sizes hold. Moreover, the distance between weight vectors increases with m.

  - Assuming that the number of weight vectors increases with m (in order to maintain the same level of coverage), the algorithm complexity increases with the number of weight vectors. It remains unclear how the number of weight vectors shall increase, e.g. polynomially or exponentially.
  - The complexity of solving an individual scalar problem grows linearly with m, and the associated problem can be formulated conveniently using the general non-linear scalarizing function proposed in this work.

- For **indicator-based MOEAs** (e.g. IBEA Zitzler and Künzli, 2004):
  - Computing the hypervolume of a solution set exactly has a complexity that is exponential in m.
  - When using Monte Carlo sampling to approximate the hypervolume combined with a constant population size, all issues mentioned above for constant population sizes hold.
  - When using a constant population size, then do not change the size of the confidence interval when approximating the hypervolume through sampling as m increases.
  - Account for both the desired accuracy of the hypervolume (width of the confidence interval) and m when deciding whether to compute the hypervolume exactly or approximately. Another interesting approach is to use recently proposed methods combining exact algorithms with Monte Carlo sampling (Tang et al., 2017; Jaszkiewicz et al., 2020).
  - When assuming a population size that increases with m, it remains unclear at this stage how the number of sampling points shall be changed (or not) to reach the same level of hypervolume approximation quality.

5.3. Future work

Although this study allowed us to make headwind in terms of both theoretical and empirical contributions to understanding and dealing more efficiently with many-objective optimization problems, there is much more that we as a community can do to advance this research area. We identified three main directions for future work to advance our work further that we discuss in the following.

Consider other problem setting and types to verify theoretical results: Our empirical study considered the impact of varying the number of objectives m using a combinatorial (binary) problem but we kept both the dimension of the design space and the level of correlation between objectives constant. It would be important to verify existing and new theoretical findings for other problem settings and other artificial and real problem types, such as continuous and mixed-integer and/or distance-based problems (Köppen et al., 2005; Fieldsend et al., 2019; Cheng et al., 2016). Gaining a more profound theoretical and empirical understanding of the impact of heterogeneous objectives (Allmendinger et al., 2015; Eichfelder et al., 2015) on many-objective optimization is critical too, together with the impact of the objectives' non-linearity and the correlation between the objectives (Verel et al., 2013).

Investigate other multi-objective concepts and algorithms: Our empirical study investigated the impact of m on different problem characteristics and multi-objective concepts. Additional empirical studies could investigate if the theoretical findings hold for different multi-objective concepts and algorithms. For example, we considered the scenario where the solutions are processed in a random order, when evaluating the complexity of dominance tests and updating the Pareto archive as a function of m, while the quality of solutions generated by MOEAs should generally improve with the running time. Thus, future work can investigate the same aspects for more realistic orders of solutions or sequences of solutions generated by real MOEAs. Moreover, it would also be important to critically evaluate the impact of m on other multi-objective concepts, such as non-dominance sorting. Consequently, the findings gained from the empirical analysis could be used to develop more efficient methods for dealing with many-objective problems. This includes also more efficient scalarization methods that account for variable domination structures and special structures of the parameter sets; exploring the use of these methods for problems where the preferences of the DM change over time would be very timely given the uncertain environment we are living in.

Expand the theoretical study: Extensive contributions can be made on advancing the theoretical grounding of multi-objective concepts and algorithms as a function m. For example, our theoretical work can be extended by deriving the probability for a solution to be non-dominated given a random population; the challenge here is to account for the various dependencies between solutions. Knowing this probability would allow to predict the expected number of non-dominated solutions in a population and hence facilitate the design of more efficient initialization strategies for many-objective problems. This line of research has been considered, for example, by Knowles and Corne (2007), Joshi and Deshpande (2014). Research related to heterogeneous objectives in a many-objective setup is still in its infancy and forms another direction of future research in terms of theory and algorithms. Given a problem with heterogeneous objectives and a portfolio of algorithms, an interesting theoretical and empirical question could be to investigate whether it is possible to use information available about the heterogeneity to predict which algorithm should be selected to solve the problem. This task can also be formulated as an algorithm selection problem (Rice, 1976).

CRediT authorship contribution statement

Richard Allmendinger: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Andrzej Jaszkiewicz:** Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Arnaud Liefooghe:** Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Christiane Tammer:** Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization.

Acknowledgments

This paper is a product of discussions initiated in the Dagstuhl Seminar 20031: Scalability in Multiobjective Optimization. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This research has been partially funded by the Polish Ministry of Education and Science, and by the French national research agency under Project ANR-16-CE23-0013-01.

Appendix. Scalarization methods

We study the relationships between a MO problem with the vector-valued objective function $\mathbf{f}$ and the feasible set X (see Definition 2.1) and a scalarization of the MO problem by translation invariant functions φ (see Section 4.3). Using these relationships, we explain the influence of the number of objectives in scalarization procedures.

In order to discuss a general scalarization technique (see also Section 4.3) for the characterization of solutions to MO problems, we assume a more general definition of Pareto optimality: We consider Pareto (weakly) optimal solutions defined by a nontrivial, closed, convex and pointed cone $C \subset \mathbb{R}^m$ (see Definition A.1). Such more general cones C are important for deriving monotonicity properties of the scalarization functions and for the characterization of properly efficient elements.

Definition A.1. Consider a MO problem where the objective function $\mathbf{f}$ is to "maximize" with respect to a nontrivial, closed, convex and pointed cone $C \subset \mathbb{R}^m$. A solution $\mathbf{x}^1 \in X$ is called a Pareto optimal solution with respect to $\mathbf{f}$ and C if:

$$\mathbf{f}(\mathbf{x}^1) \in \{\mathbf{f}(\mathbf{x}) \in Y \;:\; Y \bigcap (\mathbf{f}(\mathbf{x}) + (C \setminus \{0\})) = \emptyset\}.$$

Furthermore, $\mathbf{x}^1 \in X$ is called a Pareto weakly optimal solution with respect to $\mathbf{f}$ and C if:

$$\mathbf{f}(\mathbf{x}^1) \in \{\mathbf{f}(\mathbf{x}) \in Y \;:\; Y \bigcap (\mathbf{f}(\mathbf{x}) + \operatorname{int} C) = \emptyset\},$$

where $\operatorname{int} A$ denotes the interior of a set $A \subset \mathbb{R}^m$.

Remark. Of course, for the special case $C = \mathbb{R}^m_+ := \{y \in \mathbb{R}^m \mid \forall i \in \{1,\dots,m\} : \; y_i \geq 0\}$, a Pareto optimal solution in the sense of Definition A.1 coincides with a Pareto optimal solution in the sense of Definition 2.4.

Because of technical reasons, especially because of the calculus rules concerning $-\infty$ and $+\infty$ for extended real-valued functions, scalarization methods are usually formulated as minimization problem (see Tammer and Weidner, 2020, Chapter 3 and references therein). Taking into account that a Pareto optimal (maximal) solution with respect to $\mathbf{f}$ and C (see Definition A.1) is a Pareto optimal (minimal) solution with respect to $-\mathbf{f}$ and $-C$ and vice versa, we replace the MO problem by the equivalent problem to find Pareto optimal (minimal) solutions with respect to $-\mathbf{f}$ and $-C$. Analogously, since a Pareto weakly optimal (maximal) solution with respect to $\mathbf{f}$ and $\operatorname{int} C$ is a Pareto weakly optimal (minimal) solution with respect to $-\mathbf{f}$ and $-\operatorname{int} C$ and vice versa, we replace the MO problem by the equivalent problem to find Pareto weakly optimal (minimal) solutions with respect to $-\mathbf{f}$ and $-\operatorname{int} C$. For simplicity, we put $\mathbf{f} := -\mathbf{f}$ and denote the corresponding minimization problem as MO problem too.

A prominent scalarization method is the well known weighted Chebyshev scalarization:

$$\min_{\mathbf{x}\in X} \max_{i\in\{1,\dots,m\}} \lambda^*_i (f_i(\mathbf{x}) - w_i) \tag{A.1}$$

with $\lambda^* = (\lambda^*_1,\dots,\lambda^*_i,\dots,\lambda^*_m) \in \operatorname{int} \mathbb{R}^m_+$, $w_i \in \mathbb{R}$ $(i = 1,\dots,m)$.

Another important scalarization method is the ϵ-constraint problem (with $\epsilon = (\epsilon_1,\dots,\epsilon_m) \in \mathbb{R}^m$), see Haimes et al. (1971):

$$\begin{aligned} &\min f_j(\mathbf{x}) \\ &\text{subject to } f_i(\mathbf{x}) \leq \epsilon_i, \; i = 1,\dots,m, \; i \neq j, \\ &\qquad\qquad \mathbf{x} \in X \subseteq \mathbb{R}^n. \end{aligned} \tag{A.2}$$

We now explain that the scalarization functions involved in the problems (A.1) and (A.2) are special cases of a general non-linear translation invariant scalarization function given by Eq. (A.3).

For a nonempty closed subset A of $\mathbb{R}^m$ and an element $\ell \in -0^+A \setminus \{0\}$, we study a non-linear (along ℓ translation invariant) function $\varphi := \varphi_{A,\ell} : \mathbb{R}^m \to \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$ defined by:

$$\varphi_{A,\ell}(y) := \inf\{t \in \mathbb{R} \mid y \in t\ell + A\}. \tag{A.3}$$

Now, we consider the scalarized MO problem using the non-linear function $\varphi_{A,\ell}$ introduced in Eq. (A.3):

$$\min_{\mathbf{x}\in X} \varphi_{A,\ell}(\mathbf{f}(\mathbf{x})). \tag{MO($\varphi_{A,\ell}$)}$$

If we consider a nontrivial, closed, convex and pointed cone $C \subset \mathbb{R}^m$ and suppose $A - C \subseteq A$, then $\varphi_{A,\ell}$ in (A.3) is C-monotone (i.e., $y^1 \in y^2 + C$ implies $\varphi_{A,\ell}(y^1) \geq \varphi_{A,\ell}(y^2)$) and a solution of (MO($\varphi_{A,\ell}$)) is a weakly optimal solution with respect to $\mathbf{f}$ and C, see Göpfert et al. (2003, Theorem 2.3.1 (d) and Theorem 3.1.9).

We explain that many well known scalarization methods for multi-objective optimization problems are special cases of a scalarization by means of the non-linear (along ℓ translation invariant) function in Eq. (A.3).

(a) We show that the ϵ-constraint method can be described by means of functions of type Eq. (A.3): Let some $j \in \{1,\dots,m\}$ and some real values $\epsilon_i \in \mathbb{R}$, $i = 1,\dots,m, i \neq j$, be given. Then, the ϵ-constraint scalarization (see Haimes et al., 1971; Chankong and Haimes, 1983; Eichfelder, 2008; Durea et al., 2017), is given by the function $\varphi_{A,\ell}$ in Eq. (A.3) with:

$$A := \bar{b} - \mathbb{R}^m_+, \text{ with } \bar{b} = (\bar{b}_1,\dots,\bar{b}_m)^T, \bar{b}_i = \begin{cases} 0 & \text{for } i = j, \\ \epsilon_i & \text{for } i \neq j, \end{cases} \tag{A.4}$$

$$\ell = (\ell_1,\dots,\ell_m), \text{ where } \ell_i = \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{for } i \neq j. \end{cases} \tag{A.5}$$

We denote the function in Eq. (A.3) with A and ℓ as in (A.4)–(A.5) by $\varphi^{\epsilon}_{A,\ell}$. With A and ℓ given by (A.4)–(A.5), the scalarized MO problem:

$$\min_{\mathbf{x}\in X} \varphi^{\epsilon}_{A,\ell}(\mathbf{f}(\mathbf{x})) \tag{MO($\varphi^{\epsilon}_{A,\ell}$)}$$

describes the ϵ-constraint problem in (A.2).

Taking into account Göpfert et al. (2003, Theorem 2.3.1 (d) and Theorem 3.1.9), solutions of the scalarized problem in (A.2) generate weakly optimal solutions of the MO problem.

If the number m of objectives is increasing, then the number of parameters ϵ_i $(i = 1,\dots,m)$ involved in the constraints (see Eq. (A.4)) in the definition of the function $\varphi^{\epsilon}_{A,\ell}$ (i.e., the number of ϵ-constraints) is growing up. Furthermore, the supplementary ϵ-constraints (now involved in the constraints $y \in t\ell + A$ (see Eq. (A.3)) with A given by Eq. (A.4) and ℓ given by Eq. (A.5)) can destroy the structure of the MO problem's constraints such that the scalar optimization problem (MO($\varphi^{\epsilon}_{A,\ell}$)) could be more difficult to solve.

(b) Let a set B_L be given by a system of linear inequalities. We consider:

$$B_L := \{y \in \mathbb{R}^m \mid \langle a^i, y\rangle \leq \alpha_i, \; a^i \in \mathbb{R}^m, \; \alpha_i \in \mathbb{R}, \; i \in \{1,\dots,n\}\}. \tag{A.6}$$

Using a^i from this formula for B_L, we define a set $A_L \subset \mathbb{R}^m$ by:

$$A_L := \{y \in \mathbb{R}^m \mid \langle a^i, y\rangle \leq \alpha_i, \; i \in I\} \tag{A.7}$$

with the index set:

$$I := \{i \in \{1,\dots,n\} \mid \; \{y \in \mathbb{R}^m \mid \langle a^i, y\rangle = \alpha_i\} \cap B_L \cap \operatorname{int} \mathbb{R}^m_+ \neq \emptyset\}. \tag{A.8}$$

The set I is exactly the set of indices $i \in \{1,\dots,n\}$ for which the hyperplanes $\langle a^i, y\rangle = \alpha_i$ are active in the non-negative orthant. Let B_L and the corresponding set A_L defined as in Eq. (A.7), let vectors $\ell \in -0^+A_L \setminus \{0\}$ and $w \in \mathbb{R}^m$ be given. We consider the function $\varphi_{A,\ell}$ (see Eq. (A.3)) with $A = w + A_L$, i.e., we study the function $\varphi_{w+A_L,\ell} : \mathbb{R}^m \to \overline{\mathbb{R}}$ of type (A.3) given by:

$$\varphi_{w+A_L,\ell}(y) = \inf\{t \in \mathbb{R} \mid \; y \in t\ell + w + A_L\}, \qquad y \in \mathbb{R}^m. \tag{A.9}$$

The function $\varphi_{w+A_L,\ell}$ depends on the set A_L and the parameter vectors ℓ and w.

Using the non-linear function $\varphi_{w+A_L,\ell}$ given by Eq. (A.9), we consider the scalarized problem:

$$\min_{\mathbf{x}\in X} \varphi_{w+A_L,\ell}(\mathbf{f}(\mathbf{x})). \tag{MO($\varphi_{w+A_L,\ell}$)}$$

The following assertion is shown by Tammer and Winkler (2003) under more restrictive assumptions. In the following corollary, we derive our results under weaker assumptions concerning the parameter ℓ involved in the scalarizing function.

Corollary A.1. We consider the set B_L given by Eq. (A.6)) and $w \in \mathbb{R}^m$ arbitrarily chosen. Let the corresponding set A_L be given by Eqs. (A.7)–(A.8), $\ell \in -0^+A_L \setminus \{0\}$ and the function $\varphi_{w+A_L,\ell}$ given by (A.9). Assume that $\langle a^i, \ell\rangle \neq 0$ for all $i \in I$.

Then, the non-linear function $\varphi_{w+A_L,\ell}$ in Eq. (A.9) is convex and $\mathbb{R}^m_+$-monotone (i.e., $y^1 \in y^2 + \mathbb{R}^m_+$ implies $\varphi_{w+A_L,\ell}(y^1) \geq \varphi_{w+A_L,\ell}(y^2)$). Furthermore, $\varphi_{w+A_L,\ell}$ in Eq. (A.9) has the structure:

$$\varphi_{w+A_L,\ell}(y) = \max_{i\in I} \frac{\langle a^i, y\rangle - \langle a^i, w\rangle - \alpha_i}{\langle a^i, \ell\rangle}. \tag{A.10}$$

Proof. We get the assertions from Göpfert et al. (2003, Theorem 2.3.1) and:

$$\begin{aligned}\varphi_{w+A_L,\ell}(y) &= \inf\ \{t \in \mathbb{R} \mid\ y - w - t\ell \in A_L\} \\ &= \inf\ \{t \in \mathbb{R} \mid\ \langle a^i, y\rangle - \langle a^i, w\rangle - t\langle a^i, \ell\rangle \leq \alpha_i,\ i \in I\} \\ &= \inf\ \{t \in \mathbb{R} \mid\ \frac{\langle a^i, y\rangle - \langle a^i, w\rangle - \alpha_i}{\langle a^i, \ell\rangle} \leq t,\ i \in I\} \\ &= \max_{i\in I} \frac{\langle a^i, y\rangle - \langle a^i, w\rangle - \alpha_i}{\langle a^i, \ell\rangle}. \quad \square\end{aligned}$$

It is important to mention, if the number m of objectives is increasing, then the inner products in Eq. (A.10) are taken for parameter vectors w and ℓ in higher dimensional spaces $\mathbb{R}^m$. However, the scalarized problem MO($\varphi_{w+A_L,\ell}$) is not more difficult to solve because the parameter vectors w and ℓ are only involved in certain inner products.

Furthermore, we obtain corresponding results for the well known weighted Chebyshev scalarization (see the scalar problem in (A.1) with the origin as reference point w) as special case of Eq. (A.10). An important result by Kaliszewski (1994, Theorem 3.7) concerning the characterization of Pareto optimal elements of MO problems also follows from Eq. (A.10).

(c) In order to generate weakly optimal solutions of a MO problem, Pascoletti and Serafini (cf. Eichfelder, 2008, Pascoletti and Serafini, 1984) considered the following scalar surrogate problem:

$$\begin{aligned}&\min\ t \\ &\text{subject to } \mathbf{f}(\mathbf{x}) \in t\ell + a - \mathbb{R}^m_+, \\ &\qquad\qquad \mathbf{x} \in X \subseteq \mathbb{R}^n,\ t \in \mathbb{R}\end{aligned} \tag{A.11}$$

with parameter vectors $a \in \mathbb{R}^m$, $\ell \in \operatorname{int} \mathbb{R}^m_+$. The scalar problem in (A.11) is a scalarization of the MO problem by means of a function in Eq. (A.3) with $A = a - \mathbb{R}^m_+$ and $\ell \in \operatorname{int} \mathbb{R}^m_+$, namely:

$$\min_{\mathbf{x}\in X} \varphi_{a-\mathbb{R}^m_+,\ell}(\mathbf{f}(\mathbf{x})). \tag{MO($\varphi_{a-\mathbb{R}^m_+,\ell}$)}$$

Taking into account (Göpfert et al., 2003, Theorems 2.3.1 (d) and 3.1.9), solutions of (MO($\varphi_{a-\mathbb{R}^m_+,\ell}$)) generate weakly optimal solutions of the MO problem.

If the number m of objectives is increasing, then the parameter vectors $a \in \mathbb{R}^m$, $\ell \in \operatorname{int} \mathbb{R}^m_+$ belong to higher dimensional spaces. These parameter vectors are involved in the constraints in the definition of the function $\varphi_{a-\mathbb{R}^m_+,\ell}$ as well as of problem (A.11), such that the number of constraints is growing up. Furthermore, a perturbation of the structure of the constraints in the MO problem by the additional constraints is possible depending on the structure of the additional objectives such that the scalar problem in (A.11) could be more difficult to solve.

References

Aguirre, H.E., 2013. Advances on many-objective evolutionary optimization. In: Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation. ACM, pp. 641–666.

Aguirre, H., Tanaka, K., 2007. Working principles, behavior, and performance of MOEAs on MNK-landscapes. European J. Oper. Res. 181, 1670–1690.

Allmendinger, R., Handl, J., Knowles, J., 2015. Multiobjective optimization: When objectives exhibit non-uniform latencies. European J. Oper. Res. 243, 497–513.

Allmendinger, R., Knowles, J., 2013. 'Hang on a minute': Investigations on the effects of delayed objective functions in multiobjective optimization. In: International Conference on Evolutionary Multi-Criterion Optimization. Springer, pp. 6–20.

Allmendinger, R., Knowles, J., 2021. Heterogeneous objectives: State-of-the-art and future research. arXiv:2103.15546.

Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., 1999. Coherent measures of risk. Mathematical Finance 9, 203–228.

Bader, J., Zitzler, E., 2011. HypE: An algorithm for fast hypervolume-based many-objective optimization. Evol. Comput. 19, 45–76.

Bazgan, C., Jamain, F., Vanderpooten, D., 2013. On the number of non-dominated points of a multicriteria optimization problem. Discrete Appl. Math. 161, 2841–2850.

Beume, N., Fonseca, C.M., Lopez-Ibanez, M., Paquete, L., Vahrenhold, J., 2009. On the complexity of computing the hypervolume indicator. IEEE Trans. Evol. Comput. 13, 1075–1082.

Bonnisseau, J.-M., Cornet, B., 1988. Existence of equilibria when firms follow bounded losses pricing rules. J. Math. Econom. 17, 119–147.

Bonnisseau, J.-M., Crettez, B., 2007. On the characterization of efficient production vectors. Econom. Theory 31, 213–223.

Chan, T.M., 2013. Klee's measure problem made easy. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. pp. 410–419.

Chand, S., Wagner, M., 2015. Evolutionary many-objective optimization: A quick-start guide. Surv. Oper. Res. Manag. Sci. 20, 35–42.

Chankong, V., Haimes, Y.Y., 1983. Multiobjective Decision Making. In: North-Holland Series in System Science and Engineering, vol. 8, North-Holland Publishing Co., New York, p. xvii+406.

Cheng, R., Jin, Y., Olhofer, M., et al., 2016. Test problems for large-scale multiobjective and many-objective optimization. IEEE Trans. Cyber. 47, 4108–4121.

Chugh, T., Allmendinger, R., Ojalehto, V., Miettinen, K., 2018. Surrogate-assisted evolutionary biobjective optimization for objectives with non-uniform latencies. In: Proceedings of the 20th Annual Conference on Genetic and Evolutionary Computation. ACM, pp. 609–616.

Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A., et al., 2007. Evolutionary algorithms for solving multi-objective problems, vol. 5. Springer, New York.

Corne, D.W., Knowles, J.D., 2007. Techniques for highly multiobjective optimisation: Some nondominated points are better than others. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. In: GECCO '07, Association for Computing Machinery, New York, NY, USA, pp. 773–780.

Daolio, F., Liefooghe, A., Verel, S., Aguirre, H., Tanaka, K., 2017. Problem features versus algorithm performance on rugged multiobjective combinatorial fitness landscapes. Evol. Comput. 25, 555–585.

Deb, K., 2001. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, New York.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6, 182–197.

Drozdík, M., Akimoto, Y., Aguirre, H., Tanaka, K., 2015. Computational cost reduction of nondominated sorting using the M-front. IEEE Trans. Evol. Comput. 19, 659–678.

Durea, M., Strugariu, R., Tammer, C., 2017. On some methods to derive necessary and sufficient optimality conditions in vector optimization. J. Optim. Theory Appl. 175, 738–763.

Eichfelder, G., 2008. Adaptive Scalarization Methods in Multiobjective Optimization. In: Vector Optimization, Springer-Verlag, Berlin.

Eichfelder, G., Gandibleux, X., Geiger, M.J., Jahn, J., Jaszkiewicz, A., Knowles, J.D., Shukla, P.K., Trautmann, H., Wessing, S., 2015. Heterogeneous functions (WG3). In: Understanding Complexity in Multiobjective Optimization : Report from Dagstuhl Seminar 15031 / Greco, Salvatore; Klamroth, Kathrin; Knowles, Joshua D.; Rudolph, Günter. — Wadern : Schloss Dagstuhl. Vol. 5. Dagstuhl Zentrum für Informatik, pp. 121–129.

Fieldsend, J.E., 2020. Data structures for non-dominated sets: Implementations and empirical assessment of two decades of advances. In: Proceedings of the 22nd Annual Conference on Genetic and Evolutionary Computation. AMC, pp. 489–497.

Fieldsend, J.E., Chugh, T., Allmendinger, R., Miettinen, K., 2019. A feature rich distance-based many-objective visualisable test problem generator. In: Proceedings of the 21st Annual Conference on Genetic and Evolutionary Computation. ACM, pp. 541–549.

Fieldsend, J.E., Everson, R.M., Singh, S., 2003. Using unconstrained elite archives for multiobjective optimization. IEEE Trans. Evol. Comput. 7, 305–323.

Figueira, J.R., Fonseca, C.M., Halffmann, P., Klamroth, K., Paquete, L., Ruzika, S., Schulze, B., Stiglmayr, M., Willems, D., 2017. Easy to say they're hard, but hard to see they're easy-towards a categorization of tractable multiobjective combinatorial optimization problems. J. Multi-Criteria Decis. Anal. 24, 82–98.

Glasmachers, T., 2017. A fast incremental BSP tree archive for non-dominated points. In: International Conference on Evolutionary Multi-Criterion Optimization. Springer, pp. 252–266.

Göpfert, A., Riahi, H., Tammer, C., Zălinescu, C., 2003. Variational methods in partially ordered spaces. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, 17, Springer-Verlag, New York, p. xiv+350.

Guerreiro, A.P., Fonseca, C.M., 2018. Computing and updating hypervolume contributions in up to four dimensions. IEEE Trans. Evol. Comput. 22, 449–463.

Haimes, Y., Lasdon, L.S., Wismer, D.A., 1971. On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Trans. Syst. Man Cybern. 1, 296–297.

Heckendorn, R.B., Whitley, D., 1997. A walsh analysis of NK-landscapes. In: International Conference on Genetic Algorithms. Morgan Kaufmann, pp. 41–48.

Hughes, E.J., 2005. Evolutionary many-objective optimisation: Many once or one many? In: Congress on Evolutionary Computation. IEEE, pp. 222–227.

Ishibuchi, H., Tsukamoto, N., Nojima, Y., 2008. Evolutionary many-objective optimization: A short review. In: Congress on Evolutionary Computation. IEEE, pp. 2419–2426.

Ishibuchi, H., Tsukamoto, N., Sakane, Y., Nojima, Y., 2009. Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization. In: 2009 IEEE Congress on Evolutionary Computation. In: IEEE Congress on Evolutionary Computation, IEEE, pp. 530–537.

Jaschke, S., Küchler, U., 2001. Coherent risk measures and good-deal bounds. Finance and Stochastics 5, 181–200.

Jaszkiewicz, A., 2018. Improved quick hypervolume algorithm. Comput. Oper. Res. 90, 72–83.

Jaszkiewicz, A., Lust, T., 2018. ND-Tree-based update: A fast algorithm for the dynamic nondominance problem. IEEE Trans. Evol. Comput. 22, 778–791.

Jaszkiewicz, A., Susmaga, R., Zielniewicz, P., 2020. Approximate hypervolume calculation with guaranteed or confidence bounds. In: International Conference on Parallel Problem Solving from Nature. Springer, pp. 215–228.

Joshi, R., Deshpande, B., 2014. Empirical and analytical study of many-objective optimization problems: analysing distribution of nondominated solutions and population size for scalability of randomized heuristics. Memet. Comput. 6, 133–145.

Kaliszewski, I., 1994. Quantitative Pareto Analysis By Cone Separation Technique. Kluwer Academic Publishers, Boston, MA, p. xiv+160.

Kauffman, S.A., 1993. The Origins of Order. Oxford University Press.

Knowles, J., Corne, D., 2003. Properties of an adaptive archiving algorithm for storing nondominated vectors. IEEE Trans. Evol. Comput. 7, 100–116.

Knowles, J., Corne, D., 2007. Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 757–771.

Kollat, J.B., Reed, P.M., Maxwell, R., 2011. Many-objective groundwater monitoring network design using bias-aware ensemble Kalman filtering, evolutionary optimization, and visual analytics. Water Resour. Res. 47.

Köppen, M., Vicente-Garcia, R., Nickolay, B., 2005. Fuzzy-Pareto-dominance and its application in evolutionary multi-objective optimization. In: International Conference on Evolutionary Multi-Criterion Optimization. Springer, pp. 399–412.

Lacour, R., Klamroth, K., Fonseca, C.M., 2017. A box decomposition algorithm to compute the hypervolume indicator. Comput. Oper. Res. 79, 347–360.

Liefooghe, A., Daolio, F., Verel, S., Derbel, B., Aguirre, H., Tanaka, K., 2020. Landscape-aware performance prediction for evolutionary multi-objective optimization. IEEE Trans. Evol. Comput. 24, 1063–1077.

Luenberger, D.G., 1992a. Benefit functions and duality. J. Math. Econom. 21, 461–481.

Luenberger, D.G., 1992b. New optimality principles for economic efficiency and equilibrium. J. Optim. Theory Appl. 75, 221–264.

Ma, X., Zhang, Q., Tian, G., Yang, J., Zhu, Z., 2018. On tchebycheff decomposition approaches for multiobjective evolutionary optimization. IEEE Trans. Evol. Comput. 22, 226–244.

Malcolm, D.G., Roseboom, J.H., Clark, C.E., Fazar, W., 1959. Application of a technique for research and development program evaluation. Oper. Res. 7, 646–669.

Miettinen, K., 2012. Nonlinear Multiobjective Optimization, vol. 12. Springer Science & Business Media.

Mostaghim, S., Teich, J., 2005. Quad-trees: A data structure for storing Pareto sets in multiobjective evolutionary algorithms with elitism. In: Abraham, A., Jain, L., Goldberg, R. (Eds.), Evolutionary Multiobjective Optimization: Theoretical Advances and Applications. Springer London, London, pp. 81–104.

Newcombe, R., 1998. Two-sided confidence intervals for the single proportion: Comparison of seven methods. Stat. Med. 17, 857–872.

Pascoletti, A., Serafini, P., 1984. Scalarizing vector optimization problems. J. Optim. Theory Appl. 42, 499–524.

Polak, E., 1976. On the approximation of solutions to multiple criteria decision making problems. In: Zeleny, M. (Ed.), Multiple Criteria Decision Making. Springer, Berlin, pp. 271–282.

Purshouse, R.C., Fleming, P.J., 2003. Evolutionary many-objective optimisation: An exploratory analysis. In: Congress on Evolutionary Computation. 3, IEEE, pp. 2066–2073.

Rice, J.R., 1976. The algorithm selection problem. Adv. Comput. 15, 65–118.

Saaty, R.W., 1987. The analytic hierarchy process—what it is and how it is used. Math. Modelling 9, 161–176.

Scheffé, H., 1958. Experiments with mixtures. J. R. Stat. Soc. Ser. B Stat. Methodol. 20, 344–360.

Shang, K., Ishibuchi, H., He, L., Pang, L., 2021. A survey on the hypervolume indicator in evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. 25, 1–20.

Shang, K., Ishibuchi, H., Zhang, M.-L., Liu, Y., 2018. A new R2 indicator for better hypervolume approximation. In: Proceedings of the Genetic and Evolutionary Computation Conference. In: GECCO '18, Association for Computing Machinery, New York, NY, USA, pp. 745–752.

Small, B.G., McColl, B.W., Allmendinger, R., Pahle, J., López-Castejón, G., Rothwell, N.J., Knowles, J., Mendes, P., Brough, D., Kell, D.B., 2011. Efficient discovery of anti-inflammatory small-molecule combinations using evolutionary computing. Nat. Chem. Biol. 7, 902.

Sun, M., 2006. A primogenitary linked quad tree data structure and its application to discrete multiple criteria optimization. Ann. Oper. Res. 147, 87–107.

Sun, M., 2011. A primogenitary linked quad tree approach for solution storage and retrieval in heuristic binary optimization. European J. Oper. Res. 209, 228–240.

Sun, M., Steuer, R.E., 1996. Quad-trees and linear lists for identifying nondominated criterion vectors. INFORMS J. Comput. 8, 367–375.

Tammer, C., Weidner, P., 2020. Scalarization and separation by translation invariant functions. Springer.

Tammer, C., Winkler, K., 2003. A new scalarization approach and applications in multicriteria d.c. optimization. J. Nonlinear Convex Anal. 4, 365–380.

Tang, W., Liu, H., Chen, L., 2017. A fast approximate hypervolume calculation method by a novel decomposition strategy. In: Huang, D.-S., Bevilacqua, V., Premaratne, P., Gupta, P. (Eds.), Intelligent Computing Theories and Application. Springer International Publishing, Cham, pp. 14–25.

Terzijska, D., Porcelli, M., Eichfelder, G., 2014. Multi-objective optimization in the lorentz force velocimetry framework. In: International Workshop on Optimization and Inverse Problems in Electromagnetism. pp. 81–82.

Thomann, J., 2019. A trust region approach for multi-objective heterogeneous optimization. (Ph.D. thesis). Technische Universität Ilmenau, Ilmenau, Germany.

Trivedi, A., Srinivasan, D., Sanyal, K., Ghosh, A., 2017. A survey of multiobjective evolutionary algorithms based on decomposition. IEEE Trans. Evol. Comput. 21, 440–462.

Verel, S., Liefooghe, A., Jourdan, L., Dhaenens, C., 2013. On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives. European J. Oper. Res. 227, 331–342.

Wagner, T., Beume, N., Naujoks, B., 2007. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: International Conference on Evolutionary Multi-Criterion Optimization. Springer, pp. 742–756.

Wang, X., Jin, Y., Schmitt, S., Olhofer, M., 2020. Transfer learning for Gaussian process assisted evolutionary bi-objective optimization for objectives with different evaluation times. In: Proceedings of the 22nd Annual Conference on Genetic and Evolutionary Computation. ACM, pp. 587–594.

Wang, X., Jin, Y., Schmitt, S., Olhofer, M., Allmendinger, R., 2021. Transfer learning based surrogate assisted evolutionary bi-objective optimization for objectives with different evaluation times. Knowledge-Based Systems 227, 107190.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1, 67–82.

Zhang, R., Golovin, D., 2020. Random hypervolume scalarizations for provable multi-objective black box optimization. In: Proceedings of the 37th International Conference on Machine Learning. 119, PMLR, pp. 11096–11105.

Zhang, Q., Li, H., 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. 11, 712–731.

Zitzler, E., Künzli, S., 2004. Indicator-based selection in multiobjective search. In: International Conference on Parallel Problem Solving from Nature. Springer, pp. 832–842.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V., 2003. Performance assessment of multiobjective optimizers: An analysis and review. IEEE Trans. Evol. Comput. 7, 117–132.