On the Combined Impact of Population Size and Sub-problem Selection in MOEA/D

Geoffrey Pruvost[1(✉)], Bilel Derbel[1(✉)], Arnaud Liefooghe[1(✉)], Ke Li[2(✉)], and Qingfu Zhang[3(✉)]

[1] University of Lille, CRIStAL, Inria, Lille, France
{geoffrey.pruvost,bilel.derbel,arnaud.liefooghe}@univ-lille.fr
[2] University of Exeter, Exeter, UK
k.li@exeter.ac.uk
[3] City University Hong Kong, Kowloon Tong, Hong Kong
qingfu.zhang@cityu.edu.hk

Abstract. This paper intends to understand and to improve the working principle of decomposition-based multi-objective evolutionary algorithms. We review the design of the well-established MOEA/D framework to support the smooth integration of different strategies for sub-problem selection, while emphasizing the role of the population size and of the number of offspring created at each generation. By conducting a comprehensive empirical analysis on a wide range of multi- and many-objective combinatorial NK landscapes, we provide new insights into the combined effect of those parameters on the anytime performance of the underlying search process. In particular, we show that even a simple random strategy selecting sub-problems at random outperforms existing sophisticated strategies. We also study the sensitivity of such strategies with respect to the ruggedness and the objective space dimension of the target problem.

1 Introduction

Context. Evolutionary multi-objective optimization (EMO) algorithms [7] have been proved extremely effective in computing a high-quality approximation of the Pareto set, i.e., the set of solutions providing the best trade-offs among the objectives of a multi-objective combinatorial optimization problem (MCOP). Since the working principle of an evolutionary algorithm (EA) is to evolve a *population* of solutions, this population can be explicitly mapped with the target approximation set. The goal is then to improve the quality of the population, and to guide its incumbent individuals to be as close and as diverse as possible w.r.t. the (unknown) Pareto set. Existing EMO algorithms can be distinguished according to how the population is evolved. They are based on an iterative process where at each iteration: (i) some individuals (parents) from the population are selected, (ii) new individuals (offspring) are generated using variation operators (e.g., mutation, crossover) applied to the selected parents,

L. Paquete and C. Zarges (Eds.): EvoCOP 2020, LNCS 12102, pp. 131–147, 2020.
https://doi.org/10.1007/978-3-030-43680-3_9

and (iii) a replacement process updates the population with newly generated individuals. Apart from the problem-dependent variation operators, the design of selection and replacement is well-understood to be the main challenge for an efficient and effective EMO algorithm, since these interdependent steps allow to control both the convergence of the population and its diversity. In contrast with dominance- (e.g., [7]) or indicator-based (e.g., [2]) approaches, aggregation-based approaches [16] rely on the transformation of the objective values of a solution into a scalar value, that can be used for selection and replacement. In this paper, we are interested in studying the working principles of this class of algorithms, while focusing on the so-called MOEA/D (Multi-objective evolutionary algorithm based on decomposition) [11,22], which can be considered as a state-of-the-art framework.

Motivations. The MOEA/D framework is based on the decomposition of the original MCOP into a set of smaller sub-problems that are mapped to a population of individuals. In its basic variant [22], MOEA/D considers a set of single-objective sub-problems defined using a scalarizing function transforming a multi-dimensional objective vector into a scalar value w.r.t. one weight (or direction) vector in the objective space. The population is then typically structured by mapping one individual to one sub-problem targeting a different region of the objective space. Individuals from the population are evolved following a cooperative mechanism in order for each individual (i) to optimize its own sub-problem, and also (ii) to help solving its neighboring sub-problems. The population hence ends up having a good quality w.r.t. all sub-problems. Although being extremely simple and flexible, the computational flow of MOEA/D is constantly redesigned to deal with different issues. Different MOEA/D variants have been proposed so far in the literature, e.g., to study the impact of elitist replacements [19], of generational design [13], or of stable-matching based evolution [12], and other mechanisms [1]. In this paper, we are interested in the interdependence between the population size, which is implied by the number of sub-problems defined in the initial decomposition, and the internal evolution mechanisms of MOEA/D.

The population size has a deep impact on the dynamics and performance of EAs. In MOEA/D, the sub-problems target diversified and representative regions of the Pareto front. They are usually defined to spread evenly in the objective space. Depending on the shape of the (unknown) Pareto front, and on the number of objectives, one may need to define a different number of sub-problems. Since, the population is structured following the so-defined sub-problems, it is not clear how the robustness of the MOEA/D selection and replacement strategies can be impacted by a particular setting of the population size. Conversely, it is not clear what population size shall be chosen, and how to design a selection and replacement mechanism implying a high-quality approximation. Besides, the proper setting of the population size (see e.g. [6,8,20]) in is EAs can depend on the problem properties, for example in terms of solving difficulty. EMO algorithms are no exceptions. In MOEA/D, sub-problems may have different characteristics, and the selection and replacement mechanisms can be guided by such

considerations. This is for example the case for a number of MOEA/D variants where it is argued that some sub-problems might be more difficult to solve than others [18,23], and hence that the population shall be guided accordingly.

Methodology and Contribution. In this paper, we rely on the observation that the guiding principle of MOEA/D can be leveraged in order to support a simple and high-level tunable design of the selection and replacement mechanisms on one hand, while enabling a more fine-grained control over the choice of the population size, and subsequently its impact on approximation quality on the other hand. More specifically, our work can be summarized as follows:

- We consider a revised design of MOEA/D which explicitly dissociates between three components: (i) the number of individuals selected at each generation, (ii) the strategy adopted for selecting those individuals and (iii) the setting of the population size. Although some sophisticated strategies to distribute the computational effort of sub-problems exploration were integrated within some MOEA/D variants [10,18,23], to the best of our knowledge, the individual impact of such components were loosely studied in the past.
- Based on this fine-grained revised design, we conduct a comprehensive analysis about the impact of those three components on the convergence profile of MOEA/D. Our analysis is conducted in an incremental manner, with the aim of providing insights about the interdependence between those design components. In particular, we show evidence that the number of sub-problems selected at each generation plays an even more important role than the way the sub-problems are selected. Sophisticated selection strategies from the literature are shown to be outperformed by simpler, well configured strategies.
- We consider a broad range of multi- and many-objective NK landscapes, viewed as a standard and difficult family of MCOP benchmarks, which is both scalable in the number of objectives and exposes a controllable difficulty in terms of ruggedness. By a thorough benchmarking effort, we are then able to better elicit the impact of the MOEA/D population size, and the robustness of selection strategies on the (anytime) approximation quality.

It is worth noticing that our work shall not be considered as yet another variant in the MOEA/D literature. In fact, our analysis precisely aims at enlightening the main critical design parameters and components that can be hidden behind a successful MOEA/D setting. Our investigations are hence to be considered as a step towards the establishment of a more advanced component-wise configuration methodology allowing the setting up of future high-quality decomposition-based EMO algorithms for both multi- and many-objective optimization.

Outline. In Sect. 2, we recall basic definitions and we detail the working principle of MOEA/D. In Sect. 3, we describe our contribution in rethinking MOEA/D by explicitly dissociating between the population size and the number of selected sub-problems, then allowing us to leverage existing algorithms as instances of the revised framework. In Sect. 4, we present our experimental study and we

state our main findings. In Sect. 5, we conclude the paper and discuss further research.

2 Background

2.1 Multi-objective Combinatorial Optimization

A *multi-objective combinatorial optimization problem* (MCOP) can be defined by a set of M objective functions $f = (f_1, f_2, \ldots, f_M)$, and a discrete set X of feasible solutions in the *decision space*. Let $Z = f(X) \subseteq \mathbb{R}^M$ be the set of feasible outcome vectors in the *objective space*. To each solution $x \in X$ is assigned an objective vector $z \in Z$, on the basis of the vector function $f : X \to Z$. In a maximization context, an objective vector $z \in Z$ is *dominated* by a vector $z' \in Z$ iff $\forall m \in \{1, \ldots, M\}$, $z_m \leqslant z'_m$ and $\exists m \in \{1, \ldots, M\}$ s.t. $z_m < z'_m$. A solution $x \in X$ is dominated by a solution $x' \in X$ iff $f(x)$ is dominated by $f(x')$. A solution $x^\star \in X$ is *Pareto optimal* if there does not exist any other solution $x \in X$ such that $x^\star$ is dominated by x. The set of all Pareto optimal solutions is the *Pareto set*. Its mapping in the objective space is the *Pareto front*. The size of the Pareto set is typically exponential in the problem size. Our goal is to identify a good *Pareto set approximation*, for which EMO algorithms constitute a popular effective option [7]. As mentioned before, we are interested in aggregation-based methods, and especially in the MOEA/D framework which is sketched below.

2.2 The Conventional MOEA/D Framework

Aggregation-based EMO algorithms seek good-performing solutions in multiple regions of the Pareto front by *decomposing* the original multi-objective problem into a number of *scalarized* single-objective *sub-problems* [16]. In this paper, we use the Chebyshev scalarizing function: $\mathsf{g}(x, \omega) = \max_{i \in \{1,\ldots,M\}} \omega_i \cdot \left| z_i^\star - f_i(x) \right|$, where $x \in X$, $\omega = (\omega_1, \ldots, \omega_M)$ is a positive weight vector, and $z^\star = (z_1^\star, \ldots, z_M^\star)$ is a reference point such that $z_i^\star > f_i(x)\ \forall x \in X$, $i \in \{1, \ldots, M\}$.

In MOEA/D [22], sub-problems are optimized cooperatively by defining a *neighborhood relation* between sub-problems. Given a set of μ weight vectors $\mathcal{W}_\mu = (\omega^1, \ldots, \omega^\mu)$, with $\omega^j = (\omega_1^j, \ldots, \omega_M^j)$ for every $j \in \{1, \ldots, \mu\}$, defining μ sub-problems, MOEA/D maintains a population $P_\mu = (x^1, \ldots, x^\mu)$ where each individual x^j corresponds to one sub-problem. For each sub-problem $j \in \{1, \ldots, \mu\}$, a set of neighbors $\mathcal{B}_j$ is defined by considering the T closest weight vectors based on euclidean distance. All sub-problems are considered at each generation. Given a sub-problem j, two sub-problems are selected at random from $\mathcal{B}_j$, and the two corresponding solutions are considered as parents. An offspring x' is created by means of variation (e.g., crossover, mutation). For every $k \in \mathcal{B}_j$, if x' improves k's current solution x^k, then x' replaces it, i.e., if $\mathsf{g}(x', \omega^k) < \mathsf{g}(x^j, \omega^k)$ then $x^k = x'$. The algorithm loops over sub-problems, i.e., weight vectors, or equivalently over the individuals in the population, until

a stopping condition is satisfied. In the conventional MOEA/D terminology, an iteration refers to making selection, offspring generation, and replacement for *one* sub-problem. By contrast, a generation consists in processing all sub-problems once, i.e., after one generation μ offspring are generated. Notice that other issues are also addressed, such as the update of the reference point $z^\star$ required by the scalarizing function, and the option to incorporate an external archive for storing all non-dominated points found so far during the search process.

From the previous description, it should be clear that, at each iteration, MOEA/D is applying an elitist $(T+1)$-EA w.r.t. the sub-population $\mathcal{B}_i$ underlying the neighborhood of the current sub-problem. After one generation, one can roughly view MOEA/D as applying a $(\mu + \mu)$-EA w.r.t. the full population. A noticeable difference is that the basic MOEA/D is not a generational algorithm, in the sense that it does not handle the population as a whole, but rather in a local and greedy manner. This is actually a distinguishable feature of MOEA/D, since the population is structured by the initial sub-problems and evolved accordingly.

3 Revising and Leveraging the Design of MOEA/D

3.1 Positioning and Rationale

As in any EA, both the population size and the selection and replacement mechanisms of MOEA/D play a crucially important role. Firstly, a number of weight, i.e., a population size, that is too small may not only be insufficient to cover well the whole Pareto front, but may also prevent the identification of high-quality solutions for the defined sub-problems. This is because the generation of new offspring is guided by the so-implied $(T+1)$-EA for which the local sub-population of size T might be too different and hence too restrictive for generating good offspring. On the other hand, a too large population may result in a substantial waste of resources, since too many sub-problems might map to the same solution. Secondly, a small population size can be sufficient to approach the Pareto front in a reduced number of steps. However, a larger population is preferable to better cover the Pareto front. As in single-objective optimization, a larger population might also help escaping local optima [20]. As a result, it is not clear what is a proper setting of the population size in MOEA/D, since the previously discussed issues seem contradictory.

Although one can find different studies dealing with the impact of the population size in EAs [5,6,8,20], this issue is explicitly studied only to a small extent, especially for decomposition-based multi- and many-objective optimization [9,15]. For instance, in [8], offline and online scheduling strategies for controlling the population size are coupled with SMS-EMOA [2], a well-known indicator-based EMO algorithm, for bi-objective continuous benchmarks. Leveraging such a study to combinatorial domains with more than two objectives, and within the MOEA/D framework, is however a difficult question. Tightly related to the population size, other studies investigate the distribution of the computational effort over the sub-problems [3,4,10,18,23]. The rationale is that the defined sub-problems might have different degrees of difficulty and/or that the progress

over some sub-problems might be more advanced than others in the course of the search process. Hence, different adaptive mechanisms have been designed in order to detect which sub-problems to consider, or equivalently which solutions to select when generating a new offspring. A representative example of such approaches is the so-called MOEA/D–DRA (MOEA/D with dynamical resource allocation) [23], that can be considered as a state-of-the-art algorithm when dealing with the proper distribution of the computational effort over sub-problems. In MOEA/D–DRA, a utility function is defined w.r.t. the current status of sub-problems. A tournament selection is used to decide which sub-problems to select when generating a new offspring. Despite a skillful design, such an approach stays focused on the relative difficulty of solving sub-problems, while omitting to analyze the impact of the number of selected sub-problems and its interaction with both the population size and the characteristics of the underlying problem.

We propose to revise the MOEA/D framework and to study in a more explicit and systematic manner the combined effect of population size and sub-problem selection in light of the properties of the MCOP at hand. As mentioned above, this was investigated only to a small extent in the past although, as revealed by our experimental findings, it is of critical importance to reach an optimal performance when adopting the MOEA/D framework.

3.2 The Proposed MOEA/D–(μ, λ, sps) Framework

In order to better study and analyze the combined effect of population size and sub-problem selection, we propose to rely on a revised framework for MOEA/D, denoted MOEA/D–(μ, λ, sps), as defined in the high-level template depicted in Algorithm 1. This notation is inspired by the standard $(\mu+\lambda)$-EA scheme, where starting from a population of size μ, λ new individuals are generated and merged to form a new population of size μ after replacement. In the MOEA/D framework, however, this has a specific meaning as detailed in the following.

The proposed MOEA/D–(μ, λ, sps) algorithm follows the same steps as the original MOEA/D. However, it explicitly incorporates an additional component, denoted sps, which stands for the **s**ub-**p**roblem **s**election strategy. Initially, the population is generated and mapped to the initial μ weight vectors. An optional external archive is also incorporated in the usual way with no effect on the search process. The algorithm then proceeds in different generations (the outer while loop). At each generation, λ sub-problems, denoted I_λ, are selected using the sps strategy. A broad range of deterministic and stochastic selection strategies can be integrated. In particular, λ can be though as an intrinsic parameter of the EMO algorithm itself, or implied by a specific sps strategy. The so-selected sub-problems are processed in order to update the population (the inner for loop). For the purpose of this paper, we adopt the same scheme than conventional MOEA/D: selected sub-problems are processed in an iterative manner, although other generational EA schemes could be adopted. At each iteration, that is for each selected sub-problem, denoted i, some parents are selected as usual from the T-neighborhood $\mathcal{B}_i$ of weight vector ω^i w.r.t. $\mathcal{W}_\mu$. The setting of the neighborhood $\mathcal{B}_i$ can be exactly the same as in conventional MOEA/D and

Algorithm 1. High level template of MOEA/D–(μ, λ, sps)

Input: $\mathcal{W}_\mu := \{\omega^1, \ldots, \omega^\mu\}$: weights; $\mathsf{g}(\cdot \mid \omega)$: scalar function; T: neighb. size;
$\mathcal{EP} \leftarrow \varnothing$: (optional) external archive ;
$\mathcal{P}_\mu \leftarrow \{x^1, \ldots, x^\mu\}$: generate and evaluate initial population of size μ;
$z^\star \leftarrow$ initialize reference point from $\mathcal{P}_\mu$;
while *StoppingCriteria* **do**
  $I_\lambda \leftarrow \mathsf{sps}(\mathcal{W}_\mu, \mathcal{P}_\mu, history)$;
  for $i \in I_\lambda$ **do**
    $\mathcal{B}_i \leftarrow$ the T-neighborhood of sub-problem i using $\mathcal{W}_\mu$;
    $\mathcal{X} \leftarrow$ matingSelection($\mathcal{B}_i$);
    $x' \leftarrow$ variation($\mathcal{X}$);
    $F(x') \leftarrow$ evaluate x' ;
    $\mathcal{EP} \leftarrow$ update external archive using x';
    $z^\star \leftarrow$ update reference point using $F(x')$;
    $\mathcal{P}_\mu \leftarrow$ replacement($\mathcal{P}_\mu, x', \mathcal{B}_i \mid \mathsf{g}$);
    $history \leftarrow$ update search history;

its variants. However, at this step, it is important to emphasize that the considered neighborhood $\mathcal{B}_i$ is w.r.t. the whole set of available weight vectors $\mathcal{W}_\mu$, that is considering all the initially designed sub-problems, and *not only* the selected ones. In particular, $\mathcal{B}_i$ may include some sub-problems that were *not* selected by the sps strategy. This is motivated by the fact that parents that are likely to produce a good offspring should be defined w.r.t. the population as a whole, and not solely within the subset of active sub-problems at a given generation, which might be restrictive. A new offspring x' is then generated using standard variation operators (e.g., crossover, mutation). The reference point required by the scalarizing function and the optional external archive are updated. Thereafter, the offspring is considered for replacement as in the conventional MOEA/D and its variants. Here again, this is handled using the neighborhood $\mathcal{B}_i$ of the current sub-problem i, computed w.r.t. the whole population. It is worth noticing that population update is made on the basis of the scalarizing function g, which is a distinguishable feature of aggregation-based approaches.

At last, notice that we also use a *history* variable, referring to the evolution of the search state, and hence serving as a memory where any relevant information could be store for the future actions of the algorithm. In particular, we explicitly integrate the history within the sps strategy, since this will allow us to leverage some existing MOEA/D variants, as further discussed below.

3.3 Discussion and Outlook

It shall be clear from the previous description that the MOEA/D–(μ, λ, sps) framework allows us to emphasize the interdependence between three main components in a more fine-grained manner while following the same working principle than the original MOEA/D. Firstly, the number of weight vectors, or equivalently

Table 1. Different instantiations of the MOEA/D-$(\mu, \lambda, \mathsf{sps})$ framework.

Algorithm	Pop. size	# selected sub-prob.	Selection strategy	Ref.
MOEA/D	μ	μ	$\mathsf{sps}_{\text{ALL}}$	[22]
MOEA/D–DRA	μ	$\mu/5$	$\mathsf{sps}_{\text{DRA}}$	[23]
MOEA/D–RND	μ	$\lambda \leqslant \mu$	$\mathsf{sps}_{\text{RND}}$	here

the population size, is now made more explicit. In fact, the set of weight vectors now 'simply' plays the role of a global data structure to organize the individuals from the population. This structure can be used at the selection and replacement steps. In particular, one is not bound to iterate over all weight vectors, but might instead select a subset of individuals following a particular strategy. Secondly, the number of selected sub-problems λ determines directly the number of offspring to be generated at each generation. From an exploration/exploitation perspective, we believe this is of critical importance in general for $(\mu + \lambda)$-EAs, and it is now made more explicit within the MOEA/D framework. Furthermore, the λ offspring solutions are not simply generated from the individuals mapping to the selected sub-problems. Instead, parent selection interacts directly with the whole population, structured around the μ weight vectors, since the local neighborhood of each selected sub-problem may be used. Thirdly, the interaction between μ and λ is complemented more explicitly by the sub-problem selection strategy. In conventional MOEA/D for instance, the selection strategy turns out to be: $\mathsf{sps}_{\text{ALL}}$ = 'select all sub-problems', with $\lambda = \mu$. However, advanced MOEA/D variants can be captured as well. For instance, MOEA/D–DRA [23], focusing on the dynamic distribution of computations, can easily be instantiated as follows. For each sub-problem, we store and update the utility value as introduced in [23] by using the history variable. Let us recall that in MOEA/D–DRA, the utility of a sub-problem is simply the amount of progress made by solution x^i for sub-problem ω^i in terms of the scalarized fitness value $\mathsf{g}(\cdot|\omega^i)$ over different generations. In addition, M boundary weight vectors (in the objective space) are selected at each generation, and further $(\mu/5 - M)$ weight vectors are selected by means of a tournament selection of size 10. Hence, the sub-problem selection strategy turns out to be $\mathsf{sps}_{\text{DRA}}$ = 'select the boundary vectors and sub-problems using a tournament selection of size 10', with $\lambda = \mu/5$. Notice that this choice is to recall the one-fifth success rule from $(\mu + \lambda)$ evolution strategies [14].

In the reminder, MOEA/D–$(\mu, \mu, \mathsf{sps}_{\text{ALL}})$ refers to the conventional MOEA/D as described in [22], and MOEA/D–$(\mu, \mu/5, \mathsf{sps}_{\text{DRA}})$ refers to MOEA/D–DRA [23]; see Table 1. Other settings and parameters can be conveniently investigated as well. Since we are interested in the combined effect of μ, λ and sps, we also consider a simple baseline sub-problem selection strategy, denoted $\mathsf{sps}_{\text{RND}}$, which is to select a subset of sub-problems uniformly at random. Notice that our empirical analysis shall shed more lights on the behavior and the accuracy of the existing $\mathsf{sps}_{\text{DRA}}$ strategy.

4 Experimental Analysis

4.1 Experimental Setup

Multi-objective NK Landscapes. We consider multi-objective NK landscapes as a problem-independent model of multi-objective multi-modal combinatorial optimization problems [17]. Solutions are binary strings of size N and the objective vector to be maximized is defined as $f\colon \{0,1\}^N \mapsto [0,1]^M$. The parameter K defines the *ruggedness* of the problem, that is the number of (random) variables that influence the contribution of a given variable to the objectives. By increasing K from 0 to $(N-1)$, problems can be gradually tuned from smooth to rugged. We consider instances with the following settings: the problem size is set to $N = 100$, the number of objectives to $M \in \{2,3,4,5\}$, and the ruggedness to $K \in \{0,1,2,4\}$, that is, from linear to highly rugged landscapes. We generate one instance at random for each combination.

Parameter Setting. For our analysis, we consider three competing algorithms extracted from the MOEA/D–(μ, λ, sps) framework as depicted in Table 1. For the conventional MOEA/D, only one parameter is kept free, that is the population size μ. For MOEA/D–DRA, the sub-problem selection strategy is implemented as described in the original paper [23]. We further consider to experiment MOEA/D–DRA with other λ values. Recall that in the original variant, only $\mu/5$ sub-problems are selected, while including systematically the M boundary weight vectors. For fairness, we follow the same principle when implementing the MOEA/D–RND strategy. Notice that the boundary weight vectors were shown to impact the coordinates of the reference point $z^\star$ used by the scalarizing function [18]. They are then important to consider at each generation. To summarize, for both MOEA/D–DRA and MOEA/D–RND, two parameters are kept free, namely the population size μ and the number of selected sub-problems λ. They are chosen to cover a broad range of values, from very small to relatively very high, namely, $\mu \in \{1, 10, 50, 100, 500\}$ and $\lambda \in \{1, 2, 5, 10, 25, 50, 100, 150, 200, 300, 400, 450, 500\}$ such that $\lambda \leqslant \mu$.

The other common parameters are set as follows. The initial weights are generated using the methodology described in [21]. The neighborhood size is set to 20% of the population size: $T = 0.2\,\mu$. Two parents are considered for mating selection, i.e., the parent selection in the neighborhood of a current sub-problem i. The first parent is the current solution x^i, and the second one is selected uniformly at random from $\mathcal{B}_i$. Given that solutions are binary strings, we use a two-point crossover operator and a bit-flip mutation operator where each bit is flipped with a rate of $1/N$. MOEA/D–DRA involves additional parameters which are set following the recommendations from [23].

Performance Evaluation. Given the large number of parameter values (more than 2 000 different configurations), and in order to keep our experiments manageable in a reasonable amount of time, every configuration is executed 10 independent times, for a total of more than 20 000 runs. In order to appreciate the

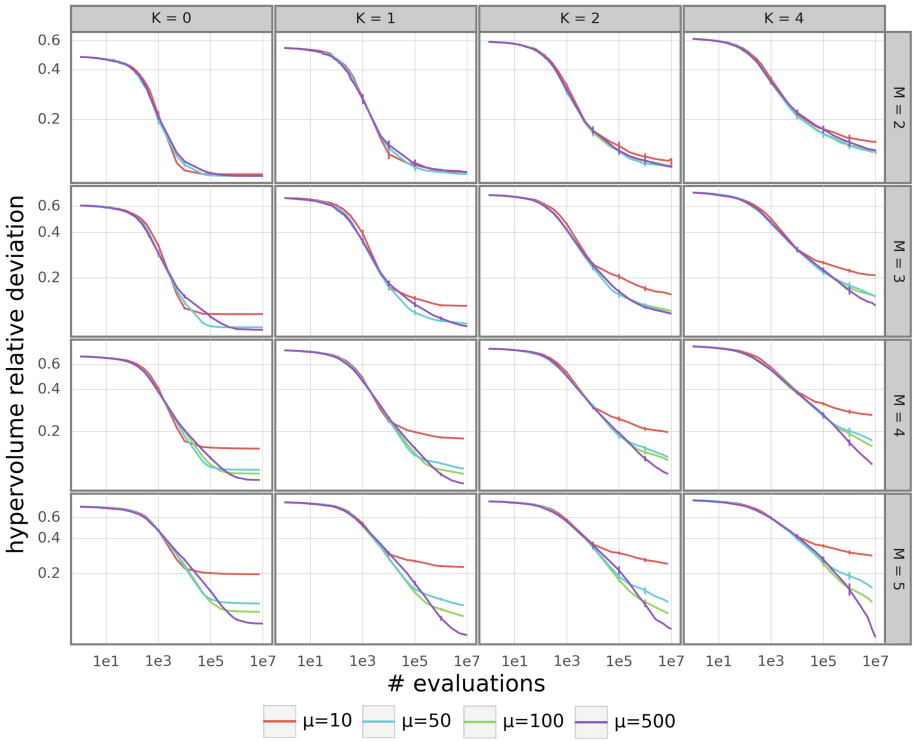

Fig. 1. Convergence profile of the conventional MOEA/D w.r.t. population size (μ).

convergence profile and the anytime behavior of the competing algorithms, we consider different stopping conditions of $\{10^0, 10^1, \ldots, 10^7\}$ calls to the evaluation function. Notice however that due to lack of space, we shall only report our findings on the basis of a representative set of our experimental data.

For performance assessment, we use the hypervolume indicator (hv) [24] to assess the quality of the obtained approximation sets, the reference point being set to the origin. More particularly, we consider the hypervolume relative deviation, computed as $\mathsf{hvrd}(A) = (\mathsf{hv}(R) - \mathsf{hv}(A))/\mathsf{hv}(R)$, where A is the obtained approximation set, and R is the best Pareto front approximation, obtained by aggregating the results over all executions and removing dominated points. As such, a lower value is better. It is important to notice that we consider the external archive, storing all non-dominated points found so far during the search process, for performance assessment. This is particularly important when comparing configurations using different population sizes.

4.2 Impact of the Population Size: $\mathsf{sps}_{\mathrm{All}}$ with Varying μ Values

We start our analysis by studying the impact of the population size for the conventional MOEA/D, that is MOEA/D–(μ, μ, $\mathsf{sps}_{\mathrm{ALL}}$) following our terminology. In

Fig. 1, we show the convergence profile using different μ values for the considered instances. Recall that hypervolume is measured on the external archive.

For a fixed budget, a smaller population size allows the search process to focus the computational effort on fewer sub-problems, hence approaching the Pareto front more quickly. By contrast, using a larger population implies more diversified solutions/sub-problems, and hence a better spreading along the Pareto front. This is typically what we observe when a small and a large budget are contrasted. In fact, a larger population size can be outperformed by a smaller one for relatively small budgets, especially when the problem is quite smooth ($K \leqslant 1$) and the number of objectives relatively high ($M \geqslant 3$). Notice also that it is not straightforward to quantify what is meant by a 'small' population, depending on the problem difficulty. For a linear bi-objective problem ($M = 2$, $K = 0$), a particularly small population size of $\mu = 10$ is sufficient to provide a relatively high accuracy. However, for quadratic many-objective problems ($M \geqslant 4$, $K = 1$), a small population size of $\mu = 10$ (resp. $\mu = 50$) is only effective up to a budget of about 10^4 (resp. 10^5) evaluations.

To summarize, it appears that the approximation quality depends both on the problem characteristics and on the available budget. For small budgets, a small population size is to be preferred. However, as the available budget grows, and as the problem difficulty increases in terms of ruggedness and number of objectives, a larger population performs better. These first observations suggest that the anytime behavior of MOEA/D can be improved by more advanced selection strategy, allowing to avoid wasting resources in processing a large number of sub-problems at each iteration, as implied by the conventional $\mathsf{sps}_{\text{ALL}}$ strategy which iterates over *all* sub-problems. This is further analyzed next.

4.3 Impact of the Sub-problem Selection Strategy

In order to fairly compare the different selection strategies, we analyze the impact of λ, i.e., the number of selected sub-problems, independently for each strategy. It is worth-noticing that *both* the value of λ and the selection strategy impact the probability of selecting a weigh vector. Our results are depicted in Fig. 2 for $\mathsf{sps}_{\text{DRA}}$ and $\mathsf{sps}_{\text{RND}}$, for different budgets and on a representative subset of instances. Other instances are not reported due to space restrictions. The main observation is that the best setting for λ depends on the considered budget, on the instance type, and on the sub-problem selection strategy itself.

Impact of λ on $\mathsf{sps}_{\text{Rnd}}$. For the random strategy $\mathsf{sps}_{\text{RND}}$ (Fig. 2, top), and for smooth problems ($K = 0$), a small λ value is found to perform better for a small budget. As the available budget grows, the λ value providing the best performance starts to increase until it reaches the population size μ. In other words, for small budgets one should select very few sub-problems at each generation, whereas for large budgets selecting all sub-problems at each generation, as done in the standard MOEA/D, appears to be a more reasonable choice. However, this tendency only holds for smooth many-objective problems. When the ruggedness increases, that is when the degree of non-linearity K grows, the effect of λ

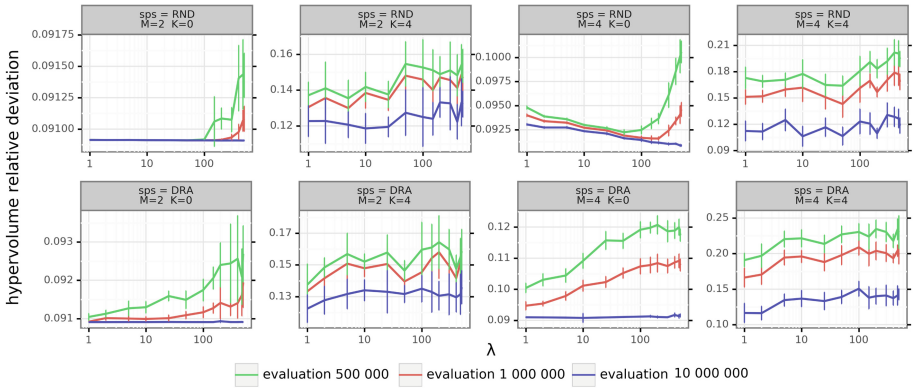

Fig. 2. Quality vs. number of selected sub-problems (λ) w.r.t. budget ($\mu = 500$).

changes. For the highest value of $K = 4$, the smallest value of $\lambda = 1$ still appears to be effective, independently of the available budget. However, the difference with a large λ value is seemingly less pronounced, especially for a relatively large budget, and the effect of λ seems to decay as the ruggedness increases. Notice also that for the 'easiest' problem instance (with $K = 0$ and $M = 2$), it is only for a small budget or for a high λ value that we observe a loss in performance. We attribute this to the fact that, when the problem is harder, search improvements are scarce within all sub-problems, it thus makes no difference to select few or many of them at each generation. By contrast, when the problem is easier, it is enough to select fewer sub-problems, as a small number of improving offspring solutions are likely sufficient to update the population.

Impact of λ on $\mathsf{sps}_{\mathrm{Dra}}$. The impact of λ appears to be different when analyzing the $\mathsf{sps}_{\mathrm{DRA}}$ strategy (Fig. 2, bottom). In fact, the effect of λ seems relatively uniform, and its optimal setting less sensitive to the available budget and instance type. More precisely, the smallest value of $\lambda = 1$ is always found to perform better, while an increasing λ value leads to a decrease in the overall approximation quality. We attribute this to the adaptive nature of $\mathsf{sps}_{\mathrm{DRA}}$, for which the probability of selecting non-interesting sub-problems is smaller for lower λ values. Interestingly, in the original setting of MOEA/D–DRA [23], from which $\mathsf{sps}_{\mathrm{DRA}}$ is extracted, the number of selected sub-problems is fixed to $\mu/5$. Not only we found that this setting can be sub-optimal, but it can actually be substantially outperformed by a simple setting of $\lambda = 1$.

$\mathsf{sps}_{\mathrm{All}}$ *vs.* $\mathsf{sps}_{\mathrm{Dra}}$ *vs.* $\mathsf{sps}_{\mathrm{Rnd}}$. Having gained insights about the effect of λ for the different selection strategies, we can fairly analyze their relative performance by using their respective optimal setting for λ. We actually show results with $\lambda = 1$ for both $\mathsf{sps}_{\mathrm{DRA}}$ and $\mathsf{sps}_{\mathrm{RND}}$. Although this setting was shown to be optimal for $\mathsf{sps}_{\mathrm{DRA}}$, it only provides a reasonably good (but sub-optimal) performance in the case of the simple random $\mathsf{sps}_{\mathrm{RND}}$ strategy, for which other λ values can be even more efficient. Our results are shown in Fig. 3 for a subset of instances.

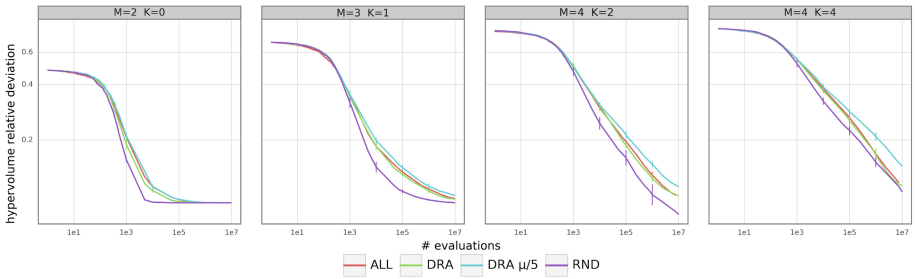

Fig. 3. Convergence profile of MOEA/D–(μ, λ, sps) w.r.t. sub-problem selection strategy ($\mu = 500$; $\lambda = 500$ for $\mathsf{sps}_{\text{ALL}}$, $\lambda \in \{1, \mu/5\}$ for $\mathsf{sps}_{\text{DRA}}$, and $\lambda = 1$ for $\mathsf{sps}_{\text{RND}}$).

Table 2. Ranks and average hvrd value (between brackets, in percentage) obtained by the different sps strategies after $\{10^4, 10^5, 10^6, 10^7\}$ evaluations (a lower value is better). Results for $\mathsf{sps}_{\text{RND}}$ and $\mathsf{sps}_{\text{DRA}}$ are for $\lambda = 1$. For each budget and instance, a rank of c indicates that the corresponding strategy was found to be significantly outperformed by c other strategies w.r.t. a Wilcoxon statistical test at a significance level of 0.05. Ranks in bold correspond to approaches that are not significantly outperformed by any other, and the underlined hvrd value corresponds to the best approach in average.

M	K	10^4 evaluations			10^5 evaluations			10^6 evaluations			10^7 evaluations		
		$\mathsf{sps}_{\text{ALL}}$	$\mathsf{sps}_{\text{DRA}}$	$\mathsf{sps}_{\text{RND}}$	$\mathsf{sps}_{\text{ALL}}$	$\mathsf{sps}_{\text{DRA}}$	$\mathsf{sps}_{\text{RND}}$	$\mathsf{sps}_{\text{ALL}}$	$\mathsf{sps}_{\text{DRA}}$	$\mathsf{sps}_{\text{RND}}$	$\mathsf{sps}_{\text{ALL}}$	$\mathsf{sps}_{\text{DRA}}$	$\mathsf{sps}_{\text{RND}}$
2	0	$2_{(11.2)}$	$1_{(10.5)}$	$\mathbf{0}_{(09.1)}$	$2_{(09.4)}$	$1_{(09.3)}$	$\mathbf{0}_{(09.0)}$	$2_{(09.1)}$	$\mathbf{0}_{(09.0)}$	$\mathbf{0}_{(09.0)}$	$\mathbf{0}_{(09.0)}$	$\mathbf{0}_{(09.0)}$	$2_{(09.0)}$
	1	$1_{(14.0)}$	$1_{(13.2)}$	$\mathbf{0}_{(11.4)}$	$1_{(10.9)}$	$1_{(10.5)}$	$\mathbf{0}_{(09.5)}$	$2_{(09.8)}$	$1_{(09.6)}$	$\mathbf{0}_{(09.2)}$	$2_{(09.5)}$	$1_{(09.4)}$	$\mathbf{0}_{(09.2)}$
	2	$1_{(17.2)}$	$\mathbf{0}_{(15.6)}$	$\mathbf{0}_{(15.4)}$	$1_{(13.0)}$	$\mathbf{0}_{(12.5)}$	$\mathbf{0}_{(11.7)}$	$\mathbf{0}_{(11.3)}$	$\mathbf{0}_{(10.9)}$	$\mathbf{0}_{(10.6)}$	$\mathbf{0}_{(10.2)}$	$\mathbf{0}_{(10.1)}$	$\mathbf{0}_{(09.8)}$
	4	$2_{(22.1)}$	$\mathbf{0}_{(19.5)}$	$\mathbf{0}_{(18.9)}$	$1_{(17.5)}$	$\mathbf{0}_{(14.9)}$	$\mathbf{0}_{(16.0)}$	$2_{(14.6)}$	$\mathbf{0}_{(13.3)}$	$\mathbf{0}_{(13.0)}$	$\mathbf{0}_{(13.1)}$	$\mathbf{0}_{(12.0)}$	$\mathbf{0}_{(12.2)}$
3	0	$1_{(15.1)}$	$1_{(15.0)}$	$\mathbf{0}_{(10.2)}$	$1_{(11.0)}$	$1_{(10.9)}$	$\mathbf{0}_{(09.2)}$	$\mathbf{0}_{(09.1)}$	$\mathbf{0}_{(09.1)}$	$\mathbf{0}_{(09.1)}$	$\mathbf{0}_{(09.0)}$	$\mathbf{0}_{(09.0)}$	$2_{(09.1)}$
	1	$1_{(18.4)}$	$1_{(18.4)}$	$\mathbf{0}_{(14.2)}$	$1_{(13.3)}$	$1_{(13.0)}$	$\mathbf{0}_{(10.5)}$	$1_{(10.8)}$	$1_{(10.5)}$	$\mathbf{0}_{(09.4)}$	$1_{(09.5)}$	$1_{(09.5)}$	$\mathbf{0}_{(09.0)}$
	2	$1_{(24.4)}$	$1_{(23.4)}$	$\mathbf{0}_{(20.6)}$	$1_{(16.3)}$	$\mathbf{0}_{(16.1)}$	$\mathbf{0}_{(14.7)}$	$2_{(12.8)}$	$1_{(12.3)}$	$\mathbf{0}_{(11.1)}$	$1_{(11.7)}$	$1_{(11.1)}$	$\mathbf{0}_{(09.4)}$
	4	$1_{(31.0)}$	$\mathbf{0}_{(29.9)}$	$\mathbf{0}_{(28.0)}$	$\mathbf{0}_{(22.7)}$	$\mathbf{0}_{(20.5)}$	$\mathbf{0}_{(21.4)}$	$\mathbf{0}_{(16.7)}$	$\mathbf{0}_{(15.4)}$	$\mathbf{0}_{(15.6)}$	$\mathbf{0}_{(13.6)}$	$\mathbf{0}_{(13.0)}$	$\mathbf{0}_{(12.2)}$
4	0	$1_{(20.5)}$	$1_{(20.5)}$	$\mathbf{0}_{(13.2)}$	$1_{(12.9)}$	$1_{(12.8)}$	$\mathbf{0}_{(09.9)}$	$\mathbf{0}_{(09.4)}$	$\mathbf{0}_{(09.4)}$	$\mathbf{0}_{(09.3)}$	$\mathbf{0}_{(09.0)}$	$\mathbf{0}_{(09.0)}$	$2_{(09.2)}$
	1	$1_{(25.0)}$	$1_{(24.9)}$	$\mathbf{0}_{(18.5)}$	$1_{(15.2)}$	$1_{(15.4)}$	$\mathbf{0}_{(11.8)}$	$1_{(09.9)}$	$1_{(09.8)}$	$\mathbf{0}_{(08.8)}$	$1_{(08.4)}$	$1_{(08.5)}$	$\mathbf{0}_{(07.9)}$
	2	$1_{(29.8)}$	$1_{(30.6)}$	$\mathbf{0}_{(24.7)}$	$1_{(19.5)}$	$1_{(18.7)}$	$\mathbf{0}_{(16.0)}$	$1_{(12.9)}$	$1_{(12.3)}$	$\mathbf{0}_{(10.1)}$	$1_{(09.8)}$	$1_{(09.9)}$	$\mathbf{0}_{(07.9)}$
	4	$1_{(38.0)}$	$1_{(37.1)}$	$\mathbf{0}_{(32.5)}$	$1_{(26.4)}$	$1_{(25.4)}$	$\mathbf{0}_{(22.5)}$	$1_{(16.8)}$	$1_{(16.6)}$	$\mathbf{0}_{(15.1)}$	$\mathbf{0}_{(11.6)}$	$\mathbf{0}_{(11.3)}$	$\mathbf{0}_{(10.5)}$
5	0	$2_{(26.3)}$	$1_{(25.2)}$	$\mathbf{0}_{(15.4)}$	$1_{(14.1)}$	$1_{(14.7)}$	$\mathbf{0}_{(10.2)}$	$\mathbf{0}_{(08.0)}$	$1_{(08.3)}$	$2_{(08.5)}$	$\mathbf{0}_{(07.4)}$	$\mathbf{0}_{(07.5)}$	$2_{(08.0)}$
	1	$1_{(29.9)}$	$1_{(29.9)}$	$\mathbf{0}_{(21.5)}$	$1_{(16.5)}$	$1_{(17.1)}$	$\mathbf{0}_{(13.0)}$	$1_{(08.3)}$	$1_{(08.5)}$	$\mathbf{0}_{(07.7)}$	$\mathbf{0}_{(05.9)}$	$\mathbf{0}_{(06.1)}$	$1_{(06.1)}$
	2	$1_{(35.2)}$	$1_{(34.1)}$	$\mathbf{0}_{(28.1)}$	$1_{(21.4)}$	$\mathbf{0}_{(20.0)}$	$\mathbf{0}_{(17.6)}$	$\mathbf{0}_{(10.9)}$	$\mathbf{0}_{(10.5)}$	$\mathbf{0}_{(09.7)}$	$\mathbf{0}_{(06.8)}$	$\mathbf{0}_{(06.2)}$	$\mathbf{0}_{(05.3)}$
	4	$1_{(41.6)}$	$1_{(40.7)}$	$\mathbf{0}_{(35.5)}$	$1_{(26.5)}$	$1_{(26.9)}$	$\mathbf{0}_{(24.1)}$	$\mathbf{0}_{(14.7)}$	$\mathbf{0}_{(15.1)}$	$\mathbf{0}_{(14.3)}$	$\mathbf{0}_{(06.0)}$	$\mathbf{0}_{(07.4)}$	$\mathbf{0}_{(07.4)}$

The $\mathsf{sps}_{\text{ALL}}$ strategy, corresponding to the conventional MOEA/D [22], and $\mathsf{sps}_{\text{DRA}}$ with $\lambda = \mu/5$, corresponding to MOEA/D–DRA [23], are also included. We can see that the simple random selection strategy $\mathsf{sps}_{\text{RND}}$ has a substantially better anytime behavior. In other words, selecting a single sub-problem at random is likely to enable identifying a high-quality approximation set more quickly, for a wide range of budgets, and independently of the instance type.

Pushing our analysis further, the only situation where a simple random strategy is outperformed by the conventional MOEA/D or by a MOEA/D–DRA setting

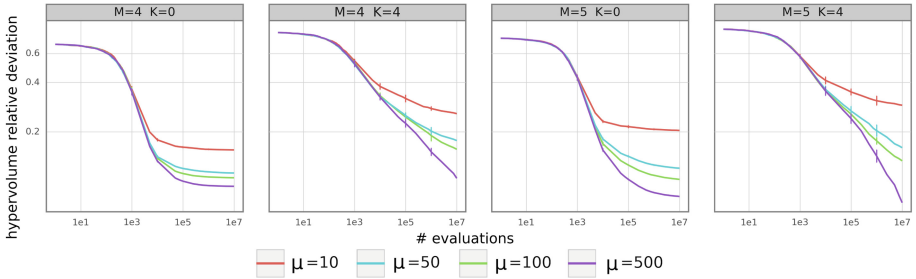

Fig. 4. Convergence profile of MOEA/D–(μ, 1, $\mathsf{sps}_{\mathrm{RND}}$) w.r.t. population size (μ).

using an optimal λ value is essentially for the very highest budget (10^7 evaluations) and when the problem is particularly smooth ($K = 0$). This can be more clearly observed in Table 2, where the relative approximation quality of the different strategies are statistically compared for different budgets. Remember however that these results are for $\lambda = 1$, which is shown to be an optimal setting for $\mathsf{sps}_{\mathrm{DRA}}$, but not necessarily for $\mathsf{sps}_{\mathrm{RND}}$ where higher λ values perform better.

4.4 Robustness of MOEA/D–(μ, λ, $\mathsf{sps}_{\mathrm{Rnd}}$) w.r.t. μ and λ

In the previous section, the population size was fixed to the highest value of $\mu = 500$. However, we have shown in Sect. 4.2 that the anytime behavior of the conventional MOEA/D can be relatively sensitive to the setting of μ, in particular for some instance types. Hence, we complement our analysis by studying the sensitivity of the $\mathsf{sps}_{\mathrm{RND}}$ strategy, which was found to have the best anytime behavior overall, w.r.t the population size μ. Results for $\mathsf{sps}_{\mathrm{RND}}$ with $\lambda = 1$ are reported in Fig. 4. In contrast with the $\mathsf{sps}_{\mathrm{ALL}}$ strategy from the conventional MOEA/D reported in Fig. 1, we can clearly see that the anytime behavior underlying $\mathsf{sps}_{\mathrm{RND}}$ is much more stable. In fact, the hypervolume increases with μ, independently of the considered budget and instance type. Notice also that when using small μ values, convergence occurs much faster for smooth problems ($K = 0$) compared against rugged ones ($K = 4$). This means that a larger population size μ, combined with a small value of λ, shall be preferred.

From a more general perspective, this observation is quite insightful since it indicates that, by increasing the number of weight vectors, one can allow for a high-level structure of the population, being eventually very large. Notice also that such a data structure can be maintained very efficiently in terms of CPU time complexity, given the scalar nature of MOEA/D. This is to contrast with, e.g., dominance-based EMO algorithms, where maintaining a large population may be computationally intensive, particularly for many-objective problems. Having such an efficient structure, the issue turns out to select some subproblems from which the (large) population is updated. A random strategy for sub-problem selection is found to work arguably well. However, in order to reach

an optimal performance, setting up the number of sub-problems λ might require further configuration issues. Overall, our analysis, reveals that a small λ value, typically ranging from 1 to 10, is recommended for relatively rugged problems, whereas a large value of λ should be preferred for smoother problems.

5 Conclusions and Perspectives

In this paper, we reviewed the design principles of the MOEA/D framework by providing a high-level, but more precise, reformulation taking inspiration from the $(\mu+\lambda)$ scheme from evolutionary computation. We analyzed the role of three design components: the population size (μ), the number of sub-problems selected (and then the number of offspring generated) at each generation (λ), and the strategy used for sub-problems selection (sps). Besides systematically informing about the combined effect of these components on the performance profile of the search process as a function of problem difficulty in terms of ruggedness and objective space dimension, our analysis opens new challenging questions on the design and practice of decomposition-based EMO algorithms.

Although we are now able to derive a parameter setting recommendation according to the general properties of the problem at hand, such properties might not always be known beforehand by the practitioner, and other properties might be considered as well. For instance, one obvious perspective would be to extend our analysis to the continuous domain. More importantly, an interesting research line would be to infer the induced landscape properties in order to learn the 'best' parameter setting, either off-line or on-line; i.e. before or during the search process. This would not only avoid the need of additional algorithm configuration (tuning) efforts, but it could also lead to an even better anytime behavior. One might for instance consider an adaptive setting where the values of μ, λ, and sps are adjusted according to the search behavior observed over different generations. Similarly, we believe that considering a problem where the objectives expose some degree of heterogeneity, e.g., in terms of solving difficulty, is worth investigating. In such a scenario, the design of an accurate sps strategy is certainly a key issue. More generally, we advocate for a more systematic analysis of such considerations for improving our fundamental understanding of the design issues behind MOEA/D and EMO algorithms in general, of the key differences between EMO algorithm classes, and of their success in solving challenging multi- and many-objective optimization problems.

Acknowledgments. This work was supported by the French national research agency (ANR-16-CE23-0013-01) and the Research Grants Council of Hong Kong (RGC Project No. A-CityU101/16).

References

1. Aghabeig, M., Jaszkiewicz, A.: Experimental analysis of design elements of scalarizing function-based multiobjective evolutionary algorithms. Soft. Comput. **23**(21), 10769–10780 (2018). https://doi.org/10.1007/s00500-018-3631-x

2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: multiobjective selection based on dominated hypervolume. Eur. J. Oper. Res. **181**(3), 1653–1669 (2007)
3. Cai, X., Li, Y., Fan, Z., Zhang, Q.: An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. IEEE Trans. Evolut. Comput. **19**(4), 508–523 (2015)
4. Chiang, T., Lai, Y.: MOEA/D-AMS: improving MOEA/D by an adaptive mating selection mechanism. In: CEC 2011, pp. 1473–1480 (2011)
5. Corus, D., Oliveto, P.S.: Standard steady state genetic algorithms can Hillclimb faster than mutation-only evolutionary algorithms. IEEE Trans. Evol. Comput. **22**(5), 720–732 (2018)
6. Črepinšek, M., Liu, S.H., Mernik, M.: Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput. Surv. **45**(3), 1–33 (2013)
7. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. Wiley, Hoboken (2001)
8. Glasmachers, T., Naujoks, B., Rudolph, G.: Start small, grow big? Saving multi-objective function evaluations. In: Bartz-Beielstein, T., Branke, J., Filipič, B., Smith, J. (eds.) PPSN 2014. LNCS, vol. 8672, pp. 579–588. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10762-2_57
9. Ishibuchi, H., Imada, R., Masuyama, N., Nojima, Y.: Two-layered weight vector specification in decomposition-based multi-objective algorithms for many-objective optimization problems. In: CEC, pp. 2434–2441 (2019)
10. Lavinas, Y., Aranha, C., Ladeira, M.: Improving resource allocation in MOEA/D with decision-space diversity metrics. In: Martín-Vide, C., Pond, G., Vega-Rodríguez, M.A. (eds.) TPNC 2019. LNCS, vol. 11934, pp. 134–146. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34500-6_9
11. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Trans. Evol. Comput. **13**(2), 284–302 (2009)
12. Li, K., Zhang, Q., Kwong, S., Li, M., Wang, R.: Stable matching-based selection in evolutionary multiobjective optimization. IEEE TEC **18**(6), 909–923 (2014)
13. Marquet, G., Derbel, B., Liefooghe, A., Talbi, E.-G.: Shake them all! Rethinking selection and replacement in MOEA/D. In: Bartz-Beielstein, T., Branke, J., Filipič, B., Smith, J. (eds.) PPSN 2014. LNCS, vol. 8672, pp. 641–651. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10762-2_63
14. Schumer, M., Steiglitz, K.: Adaptive step size random search. IEEE Trans. Autom. Control **13**(3), 270–276 (1968)
15. Tanabe, R., Ishibuchi, H.: An analysis of control parameters of MOEA/D under two different optimization scenarios. Appl. Soft Comput. **70**, 22–40 (2018)
16. Trivedi, A., Srinivasan, D., Sanyal, K., Ghosh, A.: A survey of multiobjective evolutionary algorithms based on decomposition. IEEE TEC **21**(3), 440–462 (2017)
17. Verel, S., Liefooghe, A., Jourdan, L., Dhaenens, C.: On the structure of multiobjective combinatorial search space: MNK-Landscapes with correlated objectives. Eur. J. Oper. Res. **227**(2), 331–342 (2013)
18. Wang, P., et al.: A new resource allocation strategy based on the relationship between subproblems for MOEA/D. Inf. Sci. **501**, 337–362 (2019)
19. Wang, Z., Zhang, Q., Zhou, A., Gong, M., Jiao, L.: Adaptive replacement strategies for MOEA/D. IEEE Trans. Cybern. **46**(2), 474–486 (2016)
20. Witt, C.: Population size versus runtime of a simple evolutionary algorithm. Theor. Comput. Sci. **403**(1), 104–120 (2008)
21. Zapotecas-Martínez, S., Aguirre, H., Tanaka, K., Coello, C.: On the low-discrepancy sequences and their use in MOEA/D for high-dimensional objective

spaces. In: Congress on Evolutionary Computation (CEC 2015), pp. 2835–2842 (2015)
22. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
23. Zhou, A., Zhang, Q.: Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms. IEEE Trans. Evol. Comput. **20**(1), 52–64 (2016)
24. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)