HAL
open science

Shake them all! Rethinking Selection and Replacement in MOEA/D

Gauvain Marquet, Bilel Derbel, Arnaud Liefooghe, El-Ghazali Talbi

To cite this version:

Gauvain Marquet, Bilel Derbel, Arnaud Liefooghe, El-Ghazali Talbi. Shake them all! Rethinking Selection and Replacement in MOEA/D. Parallel Problem Solving from Nature (PPSN), Sep 2014, Ljubljana, Slovenia. hal-00987800

HAL Id: hal-00987800

https://inria.hal.science/hal-00987800v1

Submitted on 6 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Shake them all!

Rethinking Selection and Replacement in MOEA/D

Gauvain Marquet Bilel Derbel Arnaud Liefooghe El-Ghazali Talbi

Inria Lille - Nord Europe, DOLPHIN project-team, France

Abstract. In this paper, we build upon the previous efforts to enhance the search ability of MOEAD (a decomposition-based algorithm), by investigating the idea of evolving the whole population simultaneously at once. We thereby propose new alternative selection and replacement strategies that can be combined in different ways within a generic and problem-independent framework. To assess the performance of our strategies, we conduct a comprehensive experimental study on bi-objective combinatorial optimization problems. More precisely, we consider ρMNK-landscapes and knapsack problems as a benchmark, and experiment a wide range of parameter configurations for MOEAD and its variants. Our analysis reveals the effectiveness of our strategies and their robustness to parameter settings. In particular, substantial improvements are obtained compared to the conventional MOEAD.

1 Introduction

Evolutionary multi-objective optimization (EMO) algorithms [1, 2] have been proved extremely effective in computing a high-quality approximation of the Pareto set, i.e., the set of solutions providing the best compromises between the multiple objectives of an optimization problem. In particular, the so-called decomposition-based (or aggregation-based) EMO is gaining in popularity as an increasing number of studies is being devoted to their development [3–5]. Recently, MOEAD [4] (Multi-Objective Evolutionary Algorithm based on Decomposition) has attracted a lot of interest; which is due to its simplicity, approximation quality, and computational efficiency. In this paper, we seek for new alternative selection mechanisms for MOEAD at the aim of enhancing its search quality; and we focus on bi-objective combinatorial problems as a case study.

Generally speaking, MOEAD builds upon the idea of decomposing the initial multi-objective optimization problem into several single-objective *sub-problems* by means of scalarizing functions [6] configured with different weight vectors. The most original part of MOEAD is to define, for each sub-problem, a *neighborhood* structure containing the set of the closest sub-problems. Then, MOEAD iterates over sub-problems and performs the three following basic steps: (i) select parents among the neighbors of the current sub-problem, (ii) generate an offspring by applying problem-specific operators, and (iii) replace neighbors' solutions if the generated offspring is better. We remark that mating selection (Step (i)) is performed exclusively among neighbors. Assuming that nearby sub-problems

have similar solutions, the neighborhood size is critical for an accurate exploration/exploitation balance. Moreover, the replacement mechanism (Step (iii)) can lead to a situation where several neighbors are replaced by the *same* offspring. This can imply a loss of diversity, and likely a loss in performance. These issues have actually been addressed in [7], where two extra modifications have been introduced when dealing with complicated continuous Pareto sets. The first modification uses an extra probability parameter allowing parents to be selected from the whole population. The second one uses an extra parameter to bound the number of neighbors that can be replaced by a newly generated offspring.

In this paper, we propose new selection strategies to enhance the search ability of MOEAD for combinatorial problems. The idea behind our strategies stems from the observation that if an offspring is allowed to replace a neighboring solution in MOEAD, this neighboring solution is then 'lost' and it has no chance to get selected for reproduction in subsequent iterations. To overcome this shortcoming, we investigate an alternative perspective in MOEAD by optimizing all sub-problems *at once*. Intuitively, every solution from the population has a more fair chance to participate in the evolution process. This allows us to propose different strategies that can be plugged in the basic version of MOEAD. Our newly proposed strategies do not distort the basic framework of MOEAD, neither do they induce a loss in generality nor do they introduce new extra parameters; while being fully compatible with the previous modifications introduced so far. Moreover, they are proven to exhibit substantial improvements in approximation quality when compared with basic MOEAD and its modifications. Our performance assessment is in fact obtained as the byproduct of a thorough experimental analysis on two bi-objective combinatorial optimization problems, namely, knapsack and ρMNK-landscapes, and by considering a broad range of configurations. In the remainder, we first recall in Sec. 2 some basic definitions as well as a brief description of MOEAD. In Sec. 3, we describe our algorithmic contribution in designing new selection and replacement strategies for MOEAD. In Sec. 4, we present the settings of our experimental study. In Sec. 5, we state our main experimental findings. Finally, we conclude the paper in Sec. 6.

2 Background

Definitions. A *multi-objective optimization problem* can be defined by an objective function vector $f = (f_1, \ldots, f_m)$ with $m \geqslant 2$, and a set X of feasible solutions in the *solution space*. In the combinatorial case, X is a discrete set. 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 function vector $f : X \to Z$ with $z = f(x)$. In a maximization context, a solution $x \in X$ is dominated by a solution $x' \in X$ iff $\forall i \in \{1, \ldots, m\}$, $f_i(x) \leqslant f_i(x')$ and $\exists i \in \{1, \ldots, m\}$ such that $f_i(x) < f_i(x')$. A solution $x^\star \in X$ is said to be *Pareto optimal* (or *non-dominated*), 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*.

Decomposition-based EMO. Contrary to existing Pareto-based EMO algorithms, like NSGA-II or SPEA2, which explicitly use the Pareto dominance relation

in their selection mechanism, decomposition-based EMO algorithms [8] rather seek a good-performing solution in multiple regions of the Pareto front by *decomposing* the original multi-objective problem into a number *scalarized* single-objective sub-problems, which can be solved independently as in MSOPS [5], or in a dependent way as in MOEAD [4]. Many different scalarizing functions have been proposed in the literature [6]. Popular examples are the weighted sum (g^{ws}) and the weighted Tchebycheff (g^{te}) functions defined below.

$$g^{ws}(x,\lambda) = \sum_{i=1}^{m} \lambda_i \cdot \left| z_i^\star - f_i(x) \right| \quad ; \quad g^{te}(x,\lambda) = \max_{i\in\{1,\ldots,m\}} \lambda_i \cdot \left| z_i^\star - f_i(x) \right|$$

where $x \in X$ is an element from the solution space, $\lambda = (\lambda_1, \ldots, \lambda_m)$ is a weighting coefficient vector such that $\lambda_i \geqslant 0$ for all i, and $z^\star = (z_1^\star, \ldots, z_m^\star)$ is a utopian point, i.e., $\forall i, \forall x, z_i^\star > f_i(x)$. Both functions are to be minimized.

MOEA/D in a Nutshell. Let g be a scalarizing function and let $(\lambda^1, \ldots, \lambda^\mu)$ be a set of μ uniformly distributed weighting coefficient vectors, corresponding to μ sub-problems to be optimized. For each sub-problem $i \in \{1, \ldots, \mu\}$, the goal is to approximate the solution x with the best scalarizing function value $g(x, \lambda^i)$. For that purpose, MOEAD maintains a population $P = (p^1, \ldots, p^\mu)$, each individual corresponding to a good-quality solution for one sub-problem. For each sub-problem $i \in \{1, \ldots, \mu\}$, a set of neighbors $\mathcal{B}(i)$ is defined with the T closest weighting coefficient vectors. To evolve the population, subproblems are optimized iteratively. At a given iteration corresponding to one sub-problem $i \in \{1, \ldots, \mu\}$, two solutions are selected at random from $\mathcal{B}(i)$, and an offspring solution x is created by means of variation operators (mutation and crossover). A problem-specific repair or improvement heuristic is potentially applied on solution x to produce x'. Then, for every sub-problem $j \in \mathcal{B}(i)$, if x' improves over j's current solution p^j then x' replaces it; i.e., if $g(x', \lambda^j) < g(p^j, \lambda^j)$ then set $p^j = x'$. The algorithm continues looping over sub-problems, optimizing them one after the other, until a stopping condition is satisfied. We shall also consider the two modifications introduced in [7] to enhance MOEAD in the context of *continuous* complicated Pareto sets. The first one allows to select a parent from the whole population with a small probability parameter $(1-\delta)$. More precisely, when dealing with a sub-problem i, its neighborhood is set to $\mathcal{B}(i)$ with probability δ, and to the whole population P with probability $(1-\delta)$. The second one limits by a parameter nr the number of times that an offspring x', created when dealing with a sub-problem i, can replace solutions in the neighborhood of i.

3 Rethinking Selection and Replacement in MOEA/D

As mentioned in the introduction, MOEAD could suffer from a lack of diversity due to the locality of its selection and replacement mechanism. We argue that this can also be caused by the fact that in MOEAD (and its modified variants), sub-problems are optimized iteratively. In fact, since parents are selected randomly from the neighborhood of the sub-problem being processed, it might happen that a solution with the potential of producing a good offspring, gets never selected for reproduction. Additionally, because a neighbor's solution might be replaced

as soon as a better offspring is found, this solution gets actually no chance to survive in the population. To increase the chance for a solution to survive in the population, we investigate the idea of evolving the whole population *simultaneously* by optimizing all subproblems in one shot and not iteratively. This idea is depicted in Algorithm 1 and discussed more thoroughly in the following.

Algorithm 1: Our proposed framework MOEAD-**xy** ($\mathbf{x}, \mathbf{y} \in \{\mathbf{s}, \mathbf{c}\}$)

Input: $\{\lambda^1, \ldots, \lambda^\mu\}$: weight vectors w.r.t sub-problems; g: a scalarizing function; $\mathcal{B}(i)$: the neighbors of sub-problem $i \in \{1, \ldots, \mu\}$; $P = \{p^1, \ldots, p^\mu\}$: the initial population.

```
1  while Stopping Condition do
2      for i ∈ {1,...,μ} do
3          if rand(0,1) < δ then B_i ← B(i) ;                    /* Neighborhood Setting */
4          else B_i ← P ;
5          if x = s then                                         /* Selfish mating selection */
6              k ← i;
7          else if x = c then                                    /* Collective mating selection */
8              k ← rand(B_i);
9          ℓ ← rand(B_i); while ℓ = k do ℓ ← rand(B_i);
10         if rand(0,1) < cr then                                /* Variation operators */
11             o^i ← crossover(p^k, p^ℓ); o^i ← mutation(o^i);
12         else o^i ← mutation(p^k);
13         if o^i is infeasible then repair(o^i);
14     for i ∈ {1,...,μ} do c_i ← 0;
15     for i ∈ {1,...,μ} do                                      /* Environmental replacement */
16         if y = s then                                         /* Selfish replacement */
17             p' ← o^i;
18             if g(p', λ^i) < g(p^i, λ^i) then p^i ← p';
19         else if y = c then                                    /* Collective replacement */
20             shuffle(B_i);
21             for j ∈ B_i do
22                 p' ← o^j;
23                 if c_j < nr then
24                     if g(p', λ^i) < g(p^i, λ^i) then p^i ← p'; c_j ← c_j + 1 ;
```

Algorithm 1 is mainly divided in two stages (lines 2 to 13 and lines 14 to 24). Contrary to MOEAD where a single offspring is generated at each iteration, our framework is basically a $(\mu+\mu)$-EA where the first stage consists in generating μ offsprings and the second stage consists in updating the whole population for the next round. The first stage corresponds to mating selection where *one* new offspring is created for *every* subproblem. Specifically, we consider two alternatives: (i) either the solution of the current subproblem is *always* selected to be a parent and hence included for variation ($\mathbf{x} = \mathbf{s}$), or (ii) parents are picked randomly from neighbors in the usual way MOEAD proceeds ($\mathbf{x} = \mathbf{c}$). Moreover, every offspring is tagged with the identifier of the subproblem where it has been created. Thus, we can identify the subproblem that originated the creation of a given offspring. Only when all subproblems are treated and all μ new offspring solutions are created, the second stage of replacement occurs. In this stage, the subproblems are processed iteratively and we again consider two alternatives: (i) either the solution of a subproblem is compared to the offspring created at this subproblem ($\mathbf{y} = \mathbf{s}$), or (ii) the solution of the current subproblem is compared to the

offspring created in neighboring subproblems ($\mathbf{y} = \mathbf{c}$). In both cases, the solution of a subproblem gets replaced if the considered offspring shows an improvement with respect to the scalarizing function of that particular subproblem.

Algorithm 1 is fully compatible with the baseline ideas of MOEAD; in particular, with the variants in [7], i.e., parameters δ and nr. Due to lack of space, we omit describing all the standard aspects that are shared with MOEAD, e.g., weights initialization, neighborhoods, update of the reference point, archiving.

To summarize, Algorithm 1 differs from MOEAD by essentially the fact that μ offsprings for all subproblems are created at each iteration. Moreover, since two alternatives are designed for mating selection and replacement, four different variants are possible: MOEAD-$\mathbf{xy}$ with $\mathbf{x}, \mathbf{y} \in \{\mathbf{s}, \mathbf{c}\}$ — $\mathbf{s}$ (resp. $\mathbf{c}$) refers to a Selfish (resp. Collective) strategy where a subproblem privileges its own solution (resp. its neighbors' solutions). It is worth to notice that some parameter combinations may not have any impact on some algorithm variants, e.g., nr does not have an impact on MOEAD-$\mathbf{ss}$ and MOEAD-$\mathbf{cs}$, neither δ on MOEAD-$\mathbf{ss}$ when $cr = 0$.

4 Experimental Setup

We analyze our approach on bi-objective ρMNK-landscapes and knapsack problems, with a broad range of instances with different structures and sizes.

ρMNK-Landscapes. The family of ρMNK-landscapes constitutes a problem-independent model used for constructing multi-objective multi-modal landscapes with objective correlation [9–11]. A bi-objective ρMNK-landscape aims at maximizing an objective function vector $f : \{0,1\}^n \rightarrow [0,1]^2$. Solutions are binary strings of size n. The parameter k defines the number of variables that influence a particular position from the bit-string (the epistatic interactions). By increasing the number of variable interactions k from 0 to $(n-1)$, landscapes can be gradually tuned from smooth to rugged. The objective correlation parameter ρ defines the degree of conflict between the objectives. The positive (resp. negative) data correlation allows to decrease (resp. increases) the degree of conflict between the objective function values. This has an impact on the cardinality of the Pareto front [9]. We investigate six random ρMNK-landscapes for each parameter combination given in Table 1.

Knapsack. The knapsack problem is one of the most studied NP-hard problem. Given a collection of n items and a set of 2 knapsacks, the $0-1$ bi-objective bi-dimensional knapsack problem seeks a subset of items subject to capacity constraint based on a *weight function* vector $w : \{0,1\}^n \rightarrow \mathbb{N}^2$, while maximizing a *profit function* vector $p : \{0,1\}^n \rightarrow \mathbb{N}^2$. More formally, it can be stated as:

$$\begin{array}{rll} \max & \sum_{j=1}^{n} p_{ij} \cdot x_j & i \in \{1,2\} \\ \text{s.t.} & \sum_{j=1}^{n} w_{ij} \cdot x_j \leqslant c_i & i \in \{1,2\} \\ & x_j \in \{0,1\} & j \in \{1,\ldots,n\} \end{array}$$

where $p_{ij} \in \mathbb{N}$ is the profit of item j on knapsack i, $w_{ij} \in \mathbb{N}$ is the weight of item j on knapsack i, and $c_i \in \mathbb{N}$ is the capacity of knapsack i. We consider the standard instances proposed in [12], with random uncorrelated profit and weight integer values from $[10, 100]$, and where capacity is set to half of the total weight of a knapsack. Thirty different random problem instances are investigated for each parameter combination given in Table 1. Moreover, we use the same advanced *weighted* repairing procedure to handle constraints as in MOEAD [4].

Table 1. Parameter setting.

		ρMNK-landscapes m = 2, n = 128	Knapsack m = 2		
		$\rho \in \{-0.7, 0.0, 0.7\}$, $K \in \{4, 8\}$	n = 250	n = 500	n = 750
pop size	μ	64, 128, 256	150	200	250
neighborhood size	T	4, 8, 16, 32	10, 20, 30		
max. number of replacements	nr	1, 2, 3, 4, ∞	2, 4, 8, 10, ∞		
neighborhood probability	δ	0.9, 1.0			
crossover rate	cr	0.0, 0.9, 1.0			
scalarizing function	g	weighted sum (g^{ws}), weighted Tchebycheff (g^{te})			
stopping condition		10^6 evaluation function calls	10^6 repair procedure calls		

Parameter Setting. Table 1 shows the parameter settings investigated in our study. We consider the effect of the population size (μ), the neighborhood size (T), the maximum number of neighboring solutions replaced (nr), the probability to select a parent outside of a neighborhood ($1-\delta$), the scalarizing function (g), and the crossover probability (cr). The stopping condition is set to 10^6 evaluation (resp. repair) calls for ρMNK-landscapes (resp. knapsack). Standard MOEAD [4, 7] is considered, together with our four variants. We use a bit-flip mutation (where each bit is independently flipped with a rate $1/n$) and one-point crossover. The crossover probability parameter (cr) allows us to appreciate the impact of the variation operator, from a pure randomized local search algorithm ($cr = 0.0$) to a conventional genetic algorithm ($cr = 1.0$). The initial population is generated randomly. An unbounded archive of all non-dominated solutions is maintained with all the approaches. All algorithms have been executed under comparable conditions and share the same base components. Overall, we tested 15918 different configurations, each one executed 30 times. Due to space limitations, we only highlight a subset of settings allowing us to state our findings.

5 Experimental Analysis

Algorithm Comparison. We start by studying the approximation quality of competing algorithms. We follow the performance assessment protocol proposed by [13] using the hypervolume difference and multiplicative epsilon indicators [14]. The hypervolume difference indicator (I_H^-) gives the difference between the portion of the objective space that is dominated by the Pareto set approximation and some reference set. The reference point is set to the worst value obtained over all approximations, and the reference set is the best-found approximation over all tested configurations. The epsilon indicator ($I_\varepsilon^\times$) gives the minimum multiplicative factor by which the approximation found by an algorithm has to be translated in the objective space to weakly dominate the reference set.

Due to space limitations, we shall not focus on eliciting the best configuration for a specific instance; but give an overview of the differences between algorithms and their robustness to different parameters. To this end, a non-exhaustive set of results is shown in Tables 2 and 3. First, notice the strong impact of the scalarizing function (g^{ws} or g^{te}) on performance and its dependency on the considered problem. Overall, MOEAD-**sc** and MOEAD-**cc** are highly competitive and exhibit the most appealing behaviors. For knapsack, these two variants perform similarly to MOEAD. This can be explained by the relative strength of the repair function, and also by the shape of the Pareto front for knapsack problems, which is relatively easy to approximate. However, for ρMNK-landscapes with

different structures and correlations, substantial improvements are reported, independently of the parameter setting. Actually, MOEAD-**ss** is also found to be competitive, but only when the crossover operator is activated. This is because MOEAD-**ss** degenerates to a multiple independent search in this case; and thus it is more likely trapped into independent local optima. At the opposite, MOEAD-**sc** and MOEAD-**cc** are able to adequately use information from neighbors, even when only a mutation (local search) operator is considered.

The previous discussion is in general valid when conducting an "anytime" analysis. This is illustrated in Fig. 1 rendering the typical convergence of competing algorithms. We see that all algorithms are able to make improvements, with MOEAD-**sc** and MOEAD-**cc** being consistently better than MOEAD. These results confirm that shaking many solutions at once can serve the approximation quality till the early stages of the search process. We also remark that MOEAD-**sc** and MOEAD-**cc** are more systematically improving upon MOEAD-**ss** for test instances having conflicting objectives; whereas MOEAD-**ss** is able to outperform its competitors as the objective correlation gets higher. Notice in fact that our strategies induce different intensification/diversification trade-offs both at the local level of every single-objective scalarized subproblem; but also at a more global level when considering the whole approximation set. When a selfish (resp. collective) mating selection is considered, the probability that a solution in the population gets selected for reproduction is 1 (resp. $1-(1-1/T)^T$). Roughly speaking, this means that all our strategies imply diversified offsprings since no solution in the current population gets replaced before exploring its potential. At the replacement stage, if a collective strategy is adopted, then the single-objective search at every subproblem is intensified since the probability that a locally improving solution can be found is higher. But this might increase the number of copies in the current approximation set. When a selfish replacement is considered, it is more likely that the number of copies is minimized; but at the price of delaying the advance of the population towards the front. For correlated objectives, and since the front is not too large, it is sufficient that only few solutions are able to approach the front in order to get good overall performance. Thus, a selfish replacement can be accurate. This is not the case for anti-correlated objectives where both the local improvements at every subproblem and the global spread of solutions is crucial. This explains the relative performance of our strategies depending on the characteristic of the tackled problem.

Impact of Parameters. From Tables 2 and 3, we can already extract some interesting observations on the impact of parameters, e.g., notice the differences between g^{ws} and g^{te}. Further observations from our data are sketched in Fig. 2, where only MOEAD, MOEAD-**sc** and MOEAD-**cc** are highlighted. First (Fig. 2 left), we confirm the positive impact of small values of parameter nr [7] on the performance of MOEAD for combinatorial problems. However, it was not always clear what is the best value to choose independently of the other parameters, e.g., the recommended value of 2 is in fact accurate, but not always optimal. Also, the impact of parameter nr on our strategies is rather mitigated. Although we found that it could often bring improvements, the impact was not pronounced compared to the case of MOEAD. For neighborhood size T (Fig. 2 middle),

Table 2. Relative performance of a representative subset of configurations w.r.t I_H^- and $I_\varepsilon^\times$ and ρMNK-landscapes ($\mu = 128$, $T = 8$, $\delta = 1.0$) at termination. For each row, the numbers indicates how many algorithm configurations (over the other 15 configurations given in columns) outperforms the configuration under consideration with a statistical confidence level of 0.05 (the lower, the better).

			g^{te}								g^{ws}							
			MOEAD	MOEAD-SS		MOEAD-SC		MOEAD-CS		MOEAD-CC	MOEAD	MOEAD-SS		MOEAD-SC		MOEAD-CS		MOEAD-CC
		nr	∞	2	–	∞	2	–	∞	2	∞	2	–	∞	2	–	∞	2
cr	ρ	K	I_H^-															
1.0	-0.7	4	2	2	8	**0**	**0**	5	**0**	**0**	4	5	**0**	6	2	7	5	6
	-0.7	8	3	1	7	**0**	**0**	6	1	**0**	11	7	6	9	6	9	9	8
	0.0	4	2	1	**0**	**0**	**0**	2	**0**	**0**	2	**0**	**0**	**0**	**0**	**0**	**0**	2
	0.0	8	6	**0**	**0**	**0**	1	6	**0**	**0**	8	7	3	8	9	12	6	**0**
	0.7	4	14	10	**0**	2	2	2	2	1	1	2	**0**	2	2	2	2	1
	0.7	8	2	2	**0**	2	1	2	2	2	2	1	**0**	5	2	1	2	1
			$I_\varepsilon^\times$															
	-0.7	4	8	8	1	**0**	**0**	1	**0**	**0**	**0**	5	**0**	5	**0**	8	2	3
	-0.7	8	5	5	**0**	**0**	**0**	5	5	**0**	7	6	**0**	9	5	7	7	5
	0.0	4	2	1	**0**	**0**	**0**	1	**0**	**0**	**0**	**0**	1	1	1	1	**0**	1
	0.0	8	4	**0**	**0**	**0**	1	3	**0**	**0**	9	9	6	9	9	10	7	7
	0.7	4	15	4	**0**	2	2	2	2	**0**	2	2	**0**	2	2	2	3	**0**
	0.7	8	1	1	**0**	2	**0**	1	1	2	2	1	1	6	2	1	4	2
0.0			I_H^-															
	-0.7	4	**0**	4	15	**0**	**0**	4	**0**	**0**	6	5	13	5	5	12	6	6
	-0.7	8	1	1	15	1	**0**	6	**0**	**0**	9	6	6	6	6	10	6	8
	0.0	4	3	**0**	15	**0**	**0**	**0**	**0**	**0**	3	**0**	11	**0**	**0**	**0**	**0**	**0**
	0.0	8	**0**	**0**	15	**0**	**0**	4	**0**	**0**	6	8	6	8	5	6	6	5
	0.7	4	9	1	9	1	**0**	1	1	**0**	3	1	2	**0**	**0**	**0**	**0**	**0**
	0.7	8	**0**	1	**0**	4	**0**	1	1	1	2	1	**0**	1	1	1	4	4
			$I_\varepsilon^\times$															
	-0.7	4	6	8	10	1	1	1	1	1	6	2	**0**	4	5	10	2	4
	-0.7	8	3	5	5	1	1	5	1	1	8	5	**0**	7	5	10	5	9
	0.0	4	3	1	13	**0**	**0**	1	**0**	**0**	1	1	1	**0**	**0**	1	1	1
	0.0	8	**0**	**0**	8	**0**	**0**	**0**	**0**	**0**	8	8	6	8	8	8	8	8
	0.7	4	6	**0**	4	**0**	**0**	1	**0**	**0**	3	2	3	**0**	**0**	**0**	**0**	**0**
	0.7	8	**0**	1	**0**	3	**0**	1	**0**	**0**	2	**0**	**0**	2	2	**0**	10	6

Table 3. Relative performance of a representative subset of configurations for knapsack ($T = 20$, $\delta = 0.9$). Similar metrics than in Table. 2 are reported.

		g^{te}								g^{ws}							
		MOEAD	MOEAD-SS		MOEAD-SC		MOEAD-CS		MOEAD-CC	MOEAD	MOEAD-SS		MOEAD-SC		MOEAD-CS		MOEAD-CC
	nr	∞	2	–	∞	2	–	∞	2	∞	2	–	∞	2	–	∞	2
cr	*N*	I_H^-															
0.9	250	8	8	7	8	8	8	8	8	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	500	8	8	8	8	8	13	8	8	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	750	8	8	8	8	8	14	8	8	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
		$I_\varepsilon^\times$															
	250	**0**	**0**	**0**	**0**	**0**	4	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	500	8	8	7	7	7	11	7	7	**0**	5	**0**	**0**	4	1	4	5
	750	4	3	6	4	3	7	3	2	5	**0**	**0**	5	7	7	**0**	8

we found that, contrary to knapsack, small values of T are interestingly more accurate for ρMNK-landscapes. We attribute this to the influence of the crossover operator which, combined with the repair mechanism, does enable to find high-quality solutions for knapsack. However, for ρMNK-landscapes, it is more likely that the crossover fails in finding good solutions when selecting many often the parents from relatively large neighborhoods. Finally (Fig. 2 right), the impact of parameter δ for neighborhood selection is confirmed to have a positive impact on the performance. However, we find that another parameter has even more

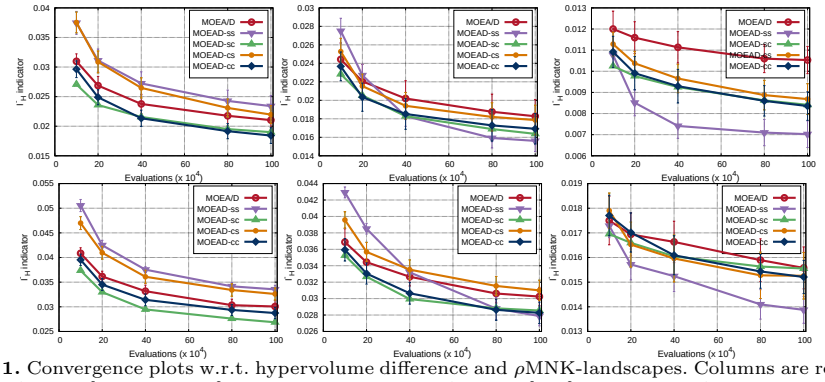

Fig. 1. Convergence plots w.r.t. hypervolume difference and ρMNK-landscapes. Columns are respectively for $\rho \in \{-0.7, 0.0, 0.7\}$. Rows are respectively for $K \in \{4, 8\}$. Results are for $\mu = 128$, $T = 8$, $\delta = 1.0$, $cr = 1.0$, $nr = 0$ and g^{te}.

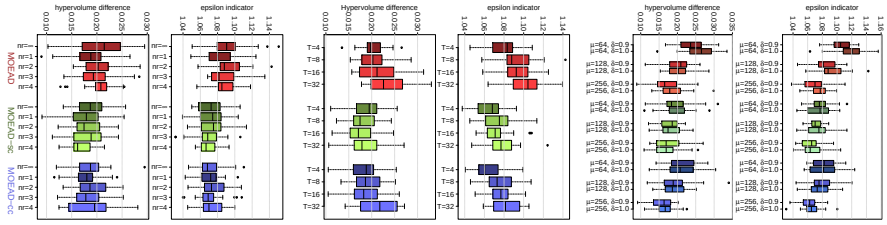

Fig. 2. Impact of parameters. The two subfigures in the left (resp. middle, right) show the impact of nr (resp. T, μ combined with δ) as shown in the vertical axis for every algorithm. The boxplots are w.r.t. the indicator depicted at the horizontal axis. Results are for a ρMNK-landscape with $\rho = -0.7$ and $K = 4$. Whenever not explicit: $\mu = 128$, $T = 8$, $cr = 1.0$, $\delta = 1.0$, $nr = 2$ and g^{te}.

effect; namely the population size μ and especially for anti-correlated instances. We attribute this to the fact that, when the Pareto front gets larger, it is more beneficial to increase the population size in order to avoid loosing promising individuals and to allow the population to be efficiently distributed.

Diversity Issues. We conclude our analysis by illustrating in Fig. 3 the size of the Pareto set approximation extracted at different iterations from the population (without the archive) for MOEAD and MOEAD-**sc**. In fact, we were able to observe that our strategy tends to maintain more spread solutions and to distribute them efficiently over the weight vectors, independently of the population size μ. We argue that this is a key feature of why our variants are able to exhibit better performance over MOEAD. Of course, this is not the only ingredient for optimal anytime performance; but it contributes much in finding a high-quality approximation, especially for conflicting objectives. This also suggests that other strategies controlling explicitly the number of clones in the whole population—treated in a global manner—may lead to an even better performance.

6 Conclusions and Perspectives

We introduced a framework incorporating four strategies to deal with selection and replacement in MOEAD. Our experimental results show that substantial improvements can be obtained. Moreover, our strategies are found to be rather robust to MOEAD configuration parameters. We argue that our study opens new possibilities for improving the design of decomposition-based algorithms

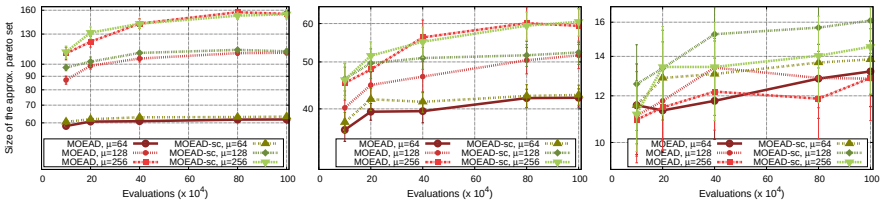

Fig. 3. Solution diversity. The y-axis gives the number of different solutions in the population (in log-scale). Results are for a ρMNK-landscape with $\rho = -0.7$ and $K = 8$ ($\mu = 128$, $T = 8$, $cr = 1.0$, $\delta = 1.0$, $nr = 2$ and g^{te}).

in several perspectives. Firstly, one can wonder whether more general $(\mu + \lambda)$-EA can be accurately embedded in our framework. Secondly, we think that our framework opens the road to high-quality local search-based MOEAD variants for combinatorial optimization problems, e.g., plugging several $(1 + \lambda)$-EAs within each subproblem. Finally, our strategies are inherently distributed in the sense that each sub-problem is optimized concurrently in parallel with the others. In this respect, an interesting research issue would be to investigate the effective parallelization of our strategies and to study their running time scalability.

References

1. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester, UK (2001)
2. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Second edn. Springer (2007)
3. Wagner, T., Beume, N., Naujoks, B.: Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization. In: EMO, Springer (2007) 742–756
4. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE TEC **11**(6) (2007) 712–731
5. Hughes, E.J.: Multiple Single Objective Pareto Sampling. In: CEC. (2003) 2678–2684
6. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer (1999)
7. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE TEC **13**(2) (2009) 284–302
8. Giagkiozis, I., Purshouse, R.C., Fleming, P.J.: Generalized Decomposition. In: EMO. (2013) 428–442
9. Verel, S., Liefooghe, A., Jourdan, L., Dhaenens, C.: On the structure of multi-objective combinatorial search space: MNK-landscapes with correlated objectives. European Journal of Operational Research **227**(2) (2013) 331–342
10. Aguirre, H.E., Tanaka, K.: Working principles, behavior, and performance of MOEAs on MNK-landscapes. EJOR **181**(3) (2007) 1670–1690
11. Kauffman, S.A.: The Origins of Order. Oxford University Press (1993)
12. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE TEC **3**(4) (1999) 257–271
13. Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK Report 214, ETH Zurich (2006)
14. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE TEC **7**(2) (2003) 117–132