



HAL
open science

Approaches for many-objective optimization: analysis and comparison on MNK-landscapes

Hernan Aguirre, Saúl Zapotecas-Martínez, Arnaud Liefooghe, Sébastien Verel,
Kiyoshi Tanaka

► **To cite this version:**

Hernan Aguirre, Saúl Zapotecas-Martínez, Arnaud Liefooghe, Sébastien Verel, Kiyoshi Tanaka. Approaches for many-objective optimization: analysis and comparison on MNK-landscapes. 13th International Conference on Artificial Evolution (EA 2015), Oct 2015, Lyon, France. pp.14-28. hal-01178613

HAL Id: hal-01178613

<https://hal.science/hal-01178613>

Submitted on 9 Sep 2021

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.

Approaches for Many-objective Optimization: Analysis and Comparison on MNK-landscapes

Hernán Aguirre¹, Saúl Zapotecas¹, Arnaud Liefooghe², Sébastien Verel³, and
Kiyoshi Tanaka¹

¹ Faculty of Engineering, Shinshu University
4-17-1 Wakasato, Nagano, 380-8553 JAPAN

² Université Lille 1 LIFL, UMR CNRS 8022, France
Inria Lille-Nord Europe, France

³ Université du Littoral Côte d'Opale, LISIC, 62228 Calais, France
{ahernan,ktanaka}@shinshu-u.ac.jp arnaud.liefooghe@lifl.fr verel@univ-littoral.fr

Abstract. This work analyses the behavior and compares the performance of MOEA/D, IBEA using the binary additive ε and the hypervolume difference indicators, and A ε S ε H as representative algorithms of decomposition, indicators, and ε -dominance based approaches for many-objective optimization. We use small MNK-landscapes to trace the dynamics of the algorithms generating high-resolution approximations of the Pareto optimal set. Also, we use large MNK-landscapes to analyze their scalability to larger search spaces.

1 Introduction

Recently, several algorithms are being proposed for many-objective optimization. Preferred approaches to implement selection in many-objective optimization are decomposition, performance indicators, and relaxations of Pareto dominance.

Decomposition based algorithms [1, 2] break down the many-objective problem into a large number of single-objective problems using scalarizing functions. The single objective problems are then solved concurrently. The scalarizing functions are usually defined in advance and remain fixed during the search. To create a set of scalarizing functions we assume a distribution of the Pareto optimal front and the algorithm aims to find good solutions that match our assumptions on distribution. Indicator based algorithms use a performance indicator function to assess the quality of a set of solutions. These algorithms optimize a single-objective function aiming to find the best subset of Pareto non-dominated solutions according to the performance indicator [3–5]. Popular indicators are additive ε , hypervolume, and R2. Relaxations of Pareto dominance modify the dominance relation to discern between initially incomparable solutions. One effective approach to relax Pareto dominance is ε -dominance [6]. ε -dominance based algorithms expand the area of dominance of some non-dominated solutions using a mapping function that depends on a parameter ε . These algorithms use ε -dominance principles to update the archive [7] or sample the instantaneous

population [8] in order to keep a subset of solutions spaced with the resolution induced by the ε mapping function. These three different approaches have led to many-objective algorithms that perform significantly better than conventional multi-objective algorithms on many-objective problems. However, there is not much work comparing them in a rigorous way and their dynamics solving many-objective problems is not yet fully understood.

This work analyses the behavior of representative algorithms that implement the above three main approaches for selection, namely the decomposition based MOEA/D, the indicator based IBEA using the binary additive ε -indicator and the binary hypervolume difference-indicator, and the ε -dominance based A ε S ε H. As reference, it also includes results by NSGA-II [9]. First, we use MNK-landscapes with 20 bits to trace the dynamics of the algorithms finding new optimal solutions and compare their performance generating high-resolution approximations of the Pareto optimal set. Then we use MNK-landscapes with 100 bits and analyze their scalability to larger search spaces. This work reveals important strengths and limitations of these algorithms for many-objective optimization, explaining their behavior and performance when convergence and diversity of the approximation is considered.

2 Algorithms

2.1 MOEA/D (Multiobjective EA Based on Decomposition)

MOEA/D [2] is a decomposition-based EMO algorithm that seeks high-quality solutions in multiple regions of the objective space by decomposing the original (multi-objective) problem into a number of scalarizing (single-objective) sub-problems. MOEA/D defines a neighboring relation among sub-problems, based on the assumption that a given sub-problem is likely to benefit from the current solutions maintained in the corresponding neighboring sub-problems. More particularly, let μ be the user-defined number of sub-problems. A set $(\lambda^1, \dots, \lambda^i, \dots, \lambda^\mu)$ of uniformly-distributed weighting coefficient vectors defines the scalarizing sub-problems, and a population $\mathcal{P} = (x^1, \dots, x^i, \dots, x^\mu)$ is maintained such that each individual x^i maps to the current solution of the corresponding sub-problem defined by λ^i . In addition, a set of neighbors $Neig(i)$ is defined by considering the T closest weighting coefficient vectors for each sub-problem i (including itself), $i \in \{1, \dots, \mu\}$. At each iteration, the population evolves with respect to a given sub-problem i . Two solutions are selected at random from $Neig(i)$ and an offspring is produced by means of crossover and mutation operators. Then, for each sub-problem $j \in Neig(i)$, the offspring x is used to replace the current solution x^j if there is an improvement in terms of the defined scalarizing function. The algorithm iterates over sub-problems until a stopping condition is satisfied.

Different scalarizing functions can be used within MOEA/D. In this paper, we use the weighted Chebyshev metric defined below.

$$g(x, \lambda) = \max_{i \in \{1, \dots, m\}} \lambda_i \cdot |z_i^* - f_i(x)| \quad (1)$$

such that x belongs to the solution space, λ is a weighting coefficient vector and z^* is a reference point.

2.2 IBEA (Indicator-Based Evolutionary Algorithm)

IBEA [3] tries to introduce a total order between solutions by means of an arbitrary binary quality indicator I . The fitness assignment scheme of IBEA is based on a pairwise comparison of solutions in a population with respect to indicator I . Each individual \mathbf{x} is assigned a fitness value measuring the “loss in quality” in the population P if \mathbf{x} was removed from it as follows

$$\text{Fitness}(\mathbf{x}) = \sum_{\mathbf{x}' \in P \setminus \{\mathbf{x}\}} (-e^{-I(\mathbf{x}', \mathbf{x})/\kappa}), \quad (2)$$

where $\kappa > 0$ is a user-defined scaling factor. Survival selection is based on an elitist strategy that combines the current population \mathcal{P}_t with its offspring \mathcal{Q}_t , iteratively deletes worst solutions until the required population size is reached, and assigns the resulting population to $\mathcal{P}_{(t+1)}$. Here, each time a solution is deleted the fitness values of the remaining individuals are updated. Parent selection for reproduction consists of binary tournaments between randomly chosen individuals using their fitness to decide the winners.

Several indicators can be used within IBEA. Here we choose to use the binary additive ϵ -indicator ($I_{\epsilon+}$) and the binary hypervolume difference-indicator (I_{HD}), as defined by the original authors [3].

$$I_{\epsilon+}(\mathbf{x}, \mathbf{x}') = \max_{i \in \{1, \dots, n\}} \{f_i(\mathbf{x}) - f_i(\mathbf{x}')\} \quad (3)$$

$$I_{HD}(\mathbf{x}, \mathbf{x}') = \begin{cases} H(\mathbf{x}') - H(\mathbf{x}) & \text{if } \mathbf{x}' \succeq \mathbf{x} \text{ or } \mathbf{x} \succeq \mathbf{x}' \\ H(\mathbf{x} + \mathbf{x}') - H(\mathbf{x}) & \text{otherwise} \end{cases} \quad (4)$$

where $\mathbf{x} \succeq \mathbf{x}'$ indicates \mathbf{x} Pareto dominates \mathbf{x}' . $I_{\epsilon+}(\mathbf{x}, \mathbf{x}')$ gives the minimum value by which a solution $\mathbf{x} \in \mathcal{P}_t$ has to, or can be translated in the objective space in order to weakly dominate another solution $\mathbf{x}' \in \mathcal{P}_t$. $H(\mathbf{x})$ give the multidimensional volume of the objective space that is dominated by \mathbf{x} . $I_{HD}(\mathbf{x}, \mathbf{x}')$ gives the hypervolume that is dominated by \mathbf{x}' but not by \mathbf{x} , $\mathbf{x}, \mathbf{x}' \in \mathcal{P}_t$. More information about IBEA can be found in [3].

2.3 The A ϵ S ϵ H

Adaptive ϵ -Sampling and ϵ -Hood (A ϵ S ϵ H) [8] is an elitist evolutionary many-objective algorithm that applies ϵ -dominance principles for survival and parent selection. There is not an explicit fitness assignment method in this algorithm.

Survival selection joins the current population \mathcal{P}_t and its offspring \mathcal{Q}_t and divide it in non-dominated fronts $\mathcal{F} = \{\mathcal{F}_i\}, i = 1, 2, \dots, N_F$ using the non-dominated sorting procedure. In the rare case where the number of non-dominated solutions is smaller than the population size $|\mathcal{F}_1| < |P|$, the sets of solutions \mathcal{F}_i

are copied iteratively to \mathcal{P}_{t+1} until it is filled; if set \mathcal{F}_i , $i > 1$, overfills \mathcal{P}_{t+1} , the required number of solutions are chosen randomly from it. On the other hand, in the common case where $|\mathcal{F}_1| > |P|$, it calls ε -*sampling* with parameter ε_s . This procedure iteratively samples randomly a solution from the set \mathcal{F}_1 , inserting the sample in \mathcal{P}_{t+1} and eliminating from \mathcal{F}_1 solutions ε -dominated by the sample. After sampling, if \mathcal{P}_{t+1} is overfilled solutions are randomly eliminated from it. Otherwise, if there is still room in \mathcal{P}_{t+1} , the required number of solutions are randomly chosen from the initially ε -dominated solutions and added to \mathcal{P}_{t+1} .

For parent selection the algorithm first uses the procedure ε -*hood creation* to cluster solutions in objective space. This procedure randomly selects an individual from the surviving population and applies ε -dominance with parameter ε_h . A neighborhood is formed by the selected solution and its ε_h -dominated solutions. Neighborhood creation is repeated until all solutions in the surviving population have been assigned to a neighborhood. Parent selection is implemented by the procedure ε -*hood mating*, which sees neighborhoods as elements of a list that are visited one at the time in a round-robin schedule. The first two parents are selected randomly from the first neighborhood in the list, the next two parents are selected randomly from the second neighborhood, and so on. When the end of the list is reached, parent selection continues with the first neighborhood in the list. Thus, all individuals have the same probability of being selected within a specified neighborhood, but due to the round-robin schedule individuals belonging to neighborhoods with fewer members have more reproduction opportunities than those belonging to neighborhoods with more members.

Both epsilon parameters ε_s and ε_h used in survival selection and parent selection, respectively, are dynamically adapted during the run of the algorithm. Further details about A ε S ε H can be found in [8].

3 Test problems, performance measures, and algorithms parameters

To evaluate the algorithms we use small and large MNK-landscapes [10] randomly generated with $M = 3, 4, 5, 6$ objectives. The small landscapes are defined with $N = 20$ bits and $K = 1$ epistatic bit (5%). We enumerate these landscapes and analyze the dynamics of the algorithms respect to the optimum set. The size of the Pareto optimal set (POS) found by enumeration and the number of non-dominated fronts are shown in Table 1 under columns $|POS|$ and *Fronts*, respectively. The same table also shows the corresponding fraction (%) of the population sizes $|P|$ to the $|POS|$ for various population sizes investigated. Also, we define large landscapes with $N = 100$ bits and $K = 5$ epistatic bits (5%) and use them to study the scalability of the algorithms to larger search spaces.

We run the algorithms for a fixed number of T generations, collecting in separate files the sets of non-dominated solutions $\mathcal{F}_1(t)$ found at each generation. The approximation of the POS for a run of the algorithm, denoted $\mathcal{A}(T)$, is built by computing the non-dominated set from all generational non-dominated sets $\mathcal{F}_1(t)$, $t = 0, 1, \dots, T$, making sure no duplicate solutions are included. In

Table 1. Size of the Pareto optimal set $|POS|$ and number of *Fronts* in the landscapes with $M = 3, 4, 5$, and 6 objectives, $N = 20$ bits, and $K = 1$ epistatic bit. Fractions $|P| / |POS|$ of population size to the size of the POS (in %) investigated in this study.

M	$ POS $	$Fronts$	$ P / POS $ (%)		
			50	100	200
3	152	258	32.9	65.8	132.6
4	1,554	76	3.2	6.4	12.9
5	6,265	29	0.8	1.6	3.2
6	16,845	22	0.3	0.6	1.2

general, the approximation at generation t is given by

$$\mathcal{X}(t) = \{\mathcal{A}(t-1) \cup \mathcal{F}_1(t) \setminus \mathcal{A}(t-1) \cap \mathcal{F}_1(t)\} \quad (5)$$

$$\mathcal{A}(t) = \{\mathbf{x} : \mathbf{x} \in \mathcal{X}(t) \wedge \nexists \mathbf{y} \in \mathcal{X}(t) \mathbf{y} \succeq \mathbf{x}\} \quad (6)$$

$$\mathcal{A}(0) = \mathcal{F}_1(0), \quad (7)$$

where $\mathbf{y} \succeq \mathbf{x}$ denotes solution \mathbf{y} Pareto dominates solution \mathbf{x} .

For small landscapes we report the basic resolution index α of the approximation at generation t [11], expressed by

$$\alpha(t) = \frac{|\{\mathbf{x} : \mathbf{x} \in \mathcal{A}(t) \wedge \mathbf{x} \in POS\}|}{|POS|}, \quad (8)$$

which gives the fraction of the accumulated number of Pareto optimal (PO) solutions found until generation t to the size of the POS. The highest resolution of the generated approximation of the POS is achieved when all Pareto optimal solutions are found. We also report three generational search assessment indices [11], the fraction τ_t^+ of Pareto optimal solutions in the population at generation t that are new respect to the previous generation, the fraction δ_t of Pareto optimal solutions dropped at generation t , and the fraction γ_t of non-dominated solutions in the population that are not Pareto optimal solutions at generation t . Table 2 summarizes these indices.

For landscapes with $N = 100$ bits, where the Pareto optimal set is unknown, we compute the non-dominated reference set \mathcal{R} from the solutions found by all algorithms. We report the Inverse Generational Distance (*IGD*) between the approximation $\mathcal{A}(T)$ found by the algorithms and the reference set \mathcal{R} . In addition, we also report the coverage C metric between the approximations $\mathcal{A}(T)$ found by the algorithms.

All algorithms use two point crossover with rate $pc = 1.0$, and bit flip mutation with rate $pm = 1/N$. In MOEA/D we use the Tchebycheff scalarizing function, as mentioned above, set the neighborhood size to 10, as suggested for knapsack problems in the original implementation of MOEA/D. The set of weights vectors is generated according to the methodology presented in [12], which projects the discrepancy given by a set of points contained in a $(k-1)$ -dimensional unit cube into a $(k-1)$ -simplex that defines the set of weights

Table 2. Generational search-assessment indices I_t . Measures are taken on non-dominated population $\mathcal{F}_1(t)$ with respect to $\mathcal{F}_1(t-1)$ and/or the POS, normalized by population size $|P|$.

I_t	Formula	Comment
τ_t^+	$ \{\mathbf{x} : \mathbf{x} \in \mathcal{F}_1(t) \wedge \mathbf{x} \notin \mathcal{F}_1(t-1) \wedge \mathbf{x} \in POS\} / P $	Possibly new PO solutions
δ_t	$ \{\mathbf{x} : \mathbf{x} \in \mathcal{F}_1(t-1) \wedge \mathbf{x} \notin \mathcal{F}_1(t) \wedge \mathbf{x} \in POS\} / P $	Dropped PO solutions
γ_t	$ \{\mathbf{x} : \mathbf{x} \in \mathcal{F}_1(t) \wedge \mathbf{x} \notin POS\} / P $	Non-dominated, not PO sol.

vectors. One advantage of using this strategy is that we can define a well-distributed set of weights vectors (in terms of low discrepancy) without depending of any constant as conventional methodologies do (see e.g. [2]) and regardless of the dimension of the weights vectors. In A ϵ S ϵ H we set the reference neighborhood size H_{size}^{Ref} to 20 individuals. The mapping function $\mathbf{f}(\mathbf{x}) \mapsto^\epsilon \mathbf{f}'(\mathbf{x})$ used for ϵ -dominance in ϵ -sampling truncation and ϵ -hood creation is additive, $f'_i = f_i + \epsilon, i = 1, 2, \dots, m$. For IBEA, we observe the behavior of the algorithm setting the scaling factor to $\kappa = 0.05$ suggested in [3] and $\kappa = 0.001$. IBEA finds considerably fewer optimal solutions if $\kappa = 0.05$. Here we report results obtained setting $\kappa = 0.001$. The algorithms run for $T = 100$ generations with population sizes $|P| = \{50, 100, 200\}$ on landscapes with $N = 20$ bits and for $T = 1000$ generation with population size $|P| = 1000$ on landscapes with $N = 100$ bits. Results analyzed here were obtained from 30 independent runs of the algorithms.

4 Experimental Results and Discussion

4.1 Small landscapes

First, we analyze the basic resolution index $\alpha(T)$ of the approximation at the end of the run, i.e. the ratio of accumulated number of PO solutions found to the size of the POS. Results for all algorithms are shown in Fig.1 for 3, 4, 5, and 6 objectives using population sizes of $\{50, 100, 200\}$. For convenience the algorithms are labeled as A, Ie, Ihv, M, and N and correspond to A ϵ S ϵ H, IBEA $I_{\epsilon+}$, IBEA I_{HD} , MOEA/D, and NSGA-II, respectively.

For $M = 3$ objectives, note that A ϵ S ϵ H finds more Pareto optimal solutions than the other algorithms for the three population sizes tried here. MOEA/D finds more Pareto optimal solutions than NSGA-II for population size 50, but the contrary is true for population sizes 100 and 200. IBEA $I_{\epsilon+}$ and I_{HD} find consistently fewer Pareto optimal solutions than the other algorithms. In $M = 3$ the ratios of population size to the size of the Pareto optimal set are $|P|/|POS| \sim \{33, 66, 133\}$ (%) for $|P| = \{50, 100, 200\}$, respectively. That is, the population size is relatively large compared to the Pareto optimal set. In this case, note that the difference in the resolution achieved by the algorithms reduces considerably as the ratio $|P|/|POS|$ increases to very large values.

On the other hand, for 4, 5 and 6 objectives, note that overall MOEA/D finds more Pareto optimal solutions than the other algorithms, followed by A ϵ S ϵ H. NSGA-II scales up badly in the number of objectives and becomes similar or

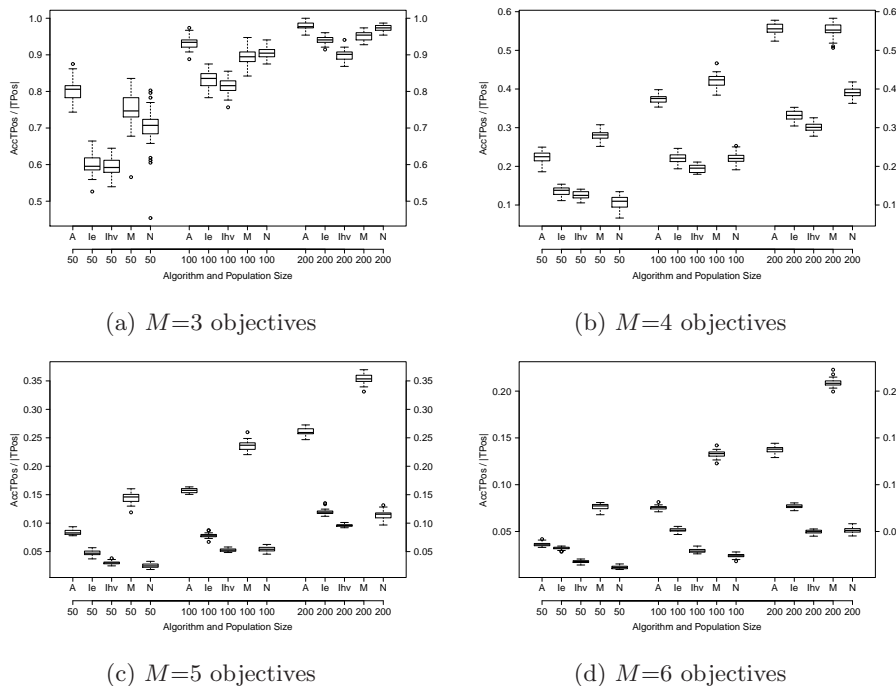


Fig. 1. Resolution of the approximation at the end of the run $\alpha(T)$, i.e. ratio of accumulated number of Pareto optimal solutions found to the size of the POS. Population sizes 50, 100, and 200 for 3, 4, 5, and 6 objectives. Algorithms AεSεH (A), IBEA $I_{\varepsilon+}$ ($I_{\varepsilon+}$), IBEA I_{HD} (I_{HD}), NSGA-II (N) and MOEA/D (M).

worse than IBEA $I_{\varepsilon+}$ and IBEA I_{HD} . In $M = 4$ the ratios are $|P|/|POS| \sim \{3.2, 6.4, 12.9\}$ (%). In this case the advantage of MOEA/D over AεSεH seen for ratios 6.4% and 3.2% disappears for the ratio 12.9% ($|P| = 200$). In $M = 5$ and $M = 6$ the ratios $|P|/|POS|$ used in our experiments are around $\{0.8, 1.6, 3.2\}$ (%) and $\{0.3, 0.6, 1.2\}$ (%). These ratios are quite small and the superiority of MOEA/D to achieve a better resolution is undisputed.

In 3, 4, and 5 objectives landscapes with $N = 20$ bits the algorithms can hit easily the Pareto optimal set after few generations. In $M = 6$ there are few optimal solutions even in the random initial population. Therefore, the above results reflect mostly the ability of the algorithms to continue discovering Pareto optimal solutions once they hit the Pareto optimal set.

In the following we analyze the dynamics of the algorithms for $M = 3$ objectives with population size $|P| = 50$, where $|P|$ is 32.9% of the $|POS|$, and for $M = 6$ with $|P| = 200$, where $|P|$ is 1.2% of the $|POS|$. Our aim is to understand the behavior of the algorithms under small and large ratios $|P|/|POS|$ and explain how the algorithms achieve the resolutions observed in Fig.1. This analysis will also help understand how the scalability to larger search spaces could be affected by the dynamics of the algorithms.

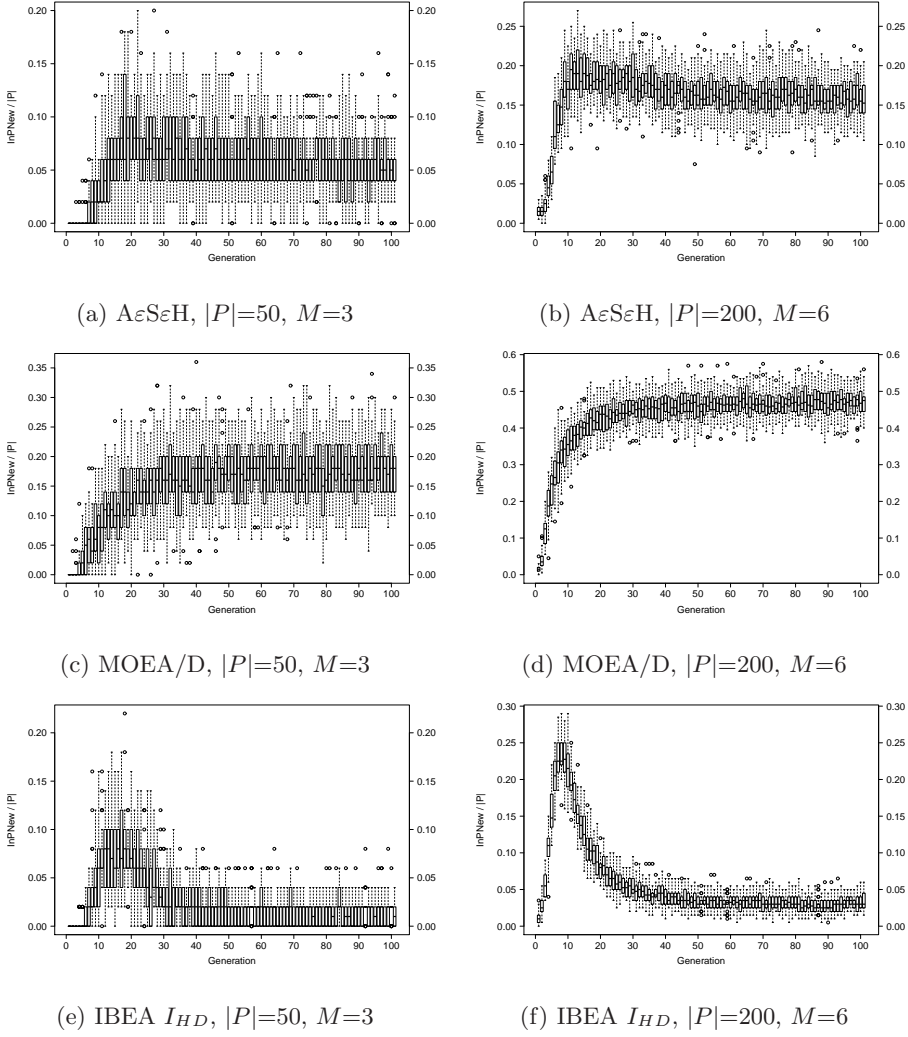


Fig. 2. Pareto optimal solutions in the population that are new respect to the previous generation. Population sizes 50 and 200 for 3 and 6 objectives, respectively. Algorithms $A\varepsilon S\varepsilon H$, MOEA/D, and IBEA I_{HD} .

Fig.2 shows the fraction τ_t^+ of Pareto optimal solutions that are new in the population respect to the previous generation. That is, τ_t^+ includes Pareto optimal solutions that are being rediscovered and also those seen for the first time. Note that τ_t^+ in $A\varepsilon S\varepsilon H$ and MOEA/D peak during the initial generations and remain close to its peak value throughout the generations. However, τ_t^+ in $A\varepsilon S\varepsilon H$ is smaller than in MOEA/D (around half), both in $M = 3$ with $|P| = 50$ (32.9% of the $|POS|$) and $M = 6$ with $|P| = 200$ (1.2% of the $|POS|$). In the case of IBEA, after τ_t^+ has reached its peak rapidly drops to a very small

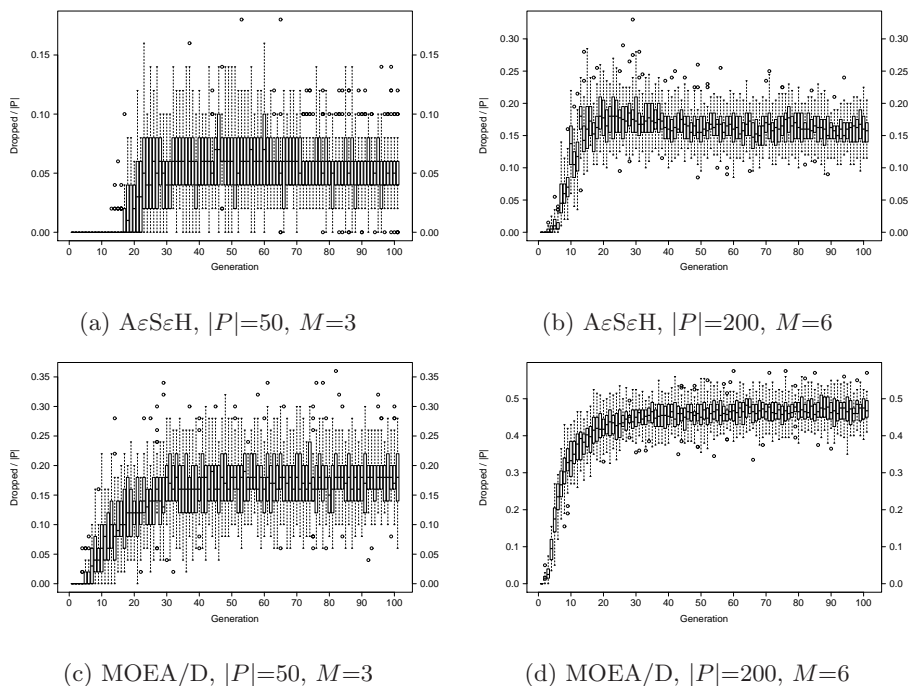


Fig. 3. Pareto optimal solutions dropped from the population. Population sizes 50 and 200 for 3 and 6 objectives, respectively. Algorithms $A\varepsilon S\varepsilon H$ and MOEA/D.

value, indicating that IBEA rediscovers and/or finds very few new Pareto optimal solutions after 30 generations.

Fig.3 shows the ratio δ_t of Pareto optimal solutions in the population that are dropped over the generations. These dropped solutions are replaced by other non-dominated solutions, optimal or not. Note that the trends of the curves are similar to those of τ_t^+ shown in Fig.2. MOEA/D drops almost three times as many Pareto optimal solutions as $A\varepsilon S\varepsilon H$ in both cases, $M = 3$ with $|P| = 50$ (32.9% of $|POS|$) and $M = 6$ with $|P| = 200$ (1.2% of $|POS|$). IBEA drops very few solutions, particularly after the algorithm has evolved few generations (results are not included here due to space limitations).

Fig.4 shows the ratio γ_t of solutions that are non-dominated in the population but are not Pareto optimal. Note that γ_t in $A\varepsilon S\varepsilon H$ is larger than in MOEA/D during the initial 20 or 10 generations, where the algorithms are approaching the optimal front and few solutions in the population are expected to be Pareto optimal. However, after this initial period, when a significant number of Pareto optimal solutions should have accumulated in the population γ_t is three times higher in MOEA/D than in $A\varepsilon S\varepsilon H$.

To summarize, whether the fraction $|P|/|POS|$ is small or large, MOEA/D discovers and rediscovers more Pareto optimal solutions than $A\varepsilon S\varepsilon H$. However, MOEA/D also drops more optimal solutions than $A\varepsilon S\varepsilon H$ and includes in its

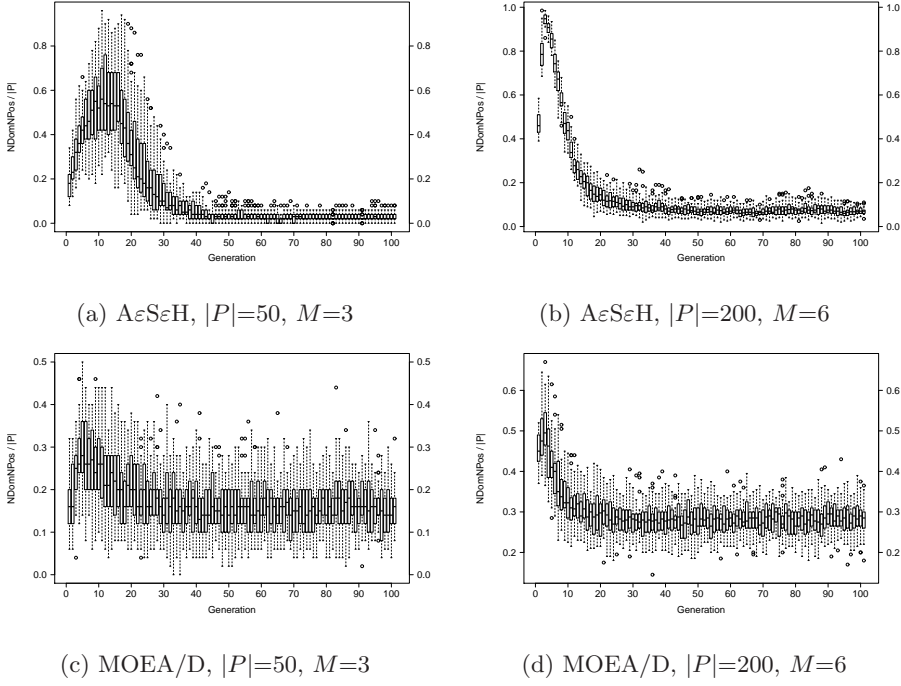


Fig. 4. Non-dominated solutions in the population that are not Pareto optimal. Population sizes 50 and 200 for 3 and 6 objectives. Algorithms $A\varepsilon S\varepsilon H$ and MOEA/D.

population a larger number of non-dominated non-Pareto optimal solutions than $A\varepsilon S\varepsilon H$. The discovery of new Pareto optimal solutions together with the ability to drop and replace them with other Pareto optimal solutions can be seen as an exploitative feature of the algorithm to continue reaching optimal solutions from optimal solutions. However, Pareto optimal solutions are also replaced with non-optimal solutions. In this case, the algorithm steps down to inferior solutions and tries to climb back again. This feature is more explorative and could help the algorithm to scape local optima, or to reach optimal solutions that cannot be reached easily from other optimal solutions. These two features are observed in both MOEA/D and $A\varepsilon S\varepsilon H$. However, the indices explored here suggest that exploration in MOEA/D is more intense than in $A\varepsilon S\varepsilon H$. The better approximation achieved by $A\varepsilon S\varepsilon H$ on 3 objectives, where there are more fronts to be climbed towards the Pareto optimal set, and the better approximations achieved by MOEA/D on larger number of objectives, where there are less fronts to be climbed, are an indication that this explorative feature could impact greatly the performance of the algorithm. In larger search spaces, it is not so simple to hit the Pareto optimal set. There, too much exploration could be detrimental to the performance of the algorithm.

An important question is how the algorithms come to drop Pareto optimal solutions from the population, particularly in favor of inferior solutions. In dom-

inance based algorithms this could happen during truncation when the number of non-dominated solutions obtained from the combined population of parents and offspring is larger than the size of the population. The scope of the Pareto relation between solutions is the population, and not all points in the landscape. Thus, solutions that appear non-dominated in the population may actually be dominated by other solutions in the landscape. For example, when the algorithm hits parts of the optimal front, even if some solutions in the combined population of parents and offspring are optimal others may be suboptimal and still appear non-dominated. In this case, Pareto optimal solutions could be dropped in favor of suboptimal solutions when the subset of surviving solutions is selected, because a dominance based algorithm cannot distinguish between non-dominated solutions. It is important to emphasize that although inferior solutions in the landscape may appear non-dominated by an optimal solution (superior solutions in general) in the population, dominance never reduces the rank of an optimal solution. In general, dominance never reduces the rank of solutions that are superior in the landscape (in the Pareto sense).

In the case of decomposition algorithms, by definition there is a different function for each sub-problem that provides a more strict order between solutions. In a combinatorial problem, the optimal solution for a sub-problem is hopefully a Pareto optimal solution. Other solutions are inferior, even if they are Pareto optimal in the multi-objective landscape. In general, from the Pareto dominance perspective, solutions that are superior in the multi-objective landscape could be ranked lower than inferior solutions. This is an important difference with dominance based approaches. When the algorithm hits the Pareto optimal set, each optimal solution in the population will be associated to one or few subproblems. These Pareto optimal solutions could be dropped in favor of a solution with higher rank in the subproblem, whether this better ranked solution is superior or not in the Pareto sense.

In the case of IBEA, the algorithm tries to introduce a total order between solutions giving higher rank to solutions located towards the ideal point. Thus, IBEA tends to converge towards the subset of solutions with highest rank located in the central region of objective space, which cardinality is the size of the population. Once there, the continuous sampling from that subset could lead to discover other Pareto optimal solutions. However, they will have a rank inferior to those in the population and thus are not eligible to replace optimal solutions. After a while, the algorithm cannot find new solutions from the same set and stagnates. Due to the total order, this algorithm includes features that can help convergence in larger subspaces, though diversity could still be an issue.

4.2 Large landscapes

In this section we present results of the algorithms on landscapes with $N = 100$ bits in order to analyze their scalability to larger search spaces. Fig.5 and Fig.6 show the inverse generational distance IGD of the approximation obtained by the algorithms and the coverage C metric between the approximations of A ϵ S ϵ H and the other algorithms, respectively. For these problems we don't know the

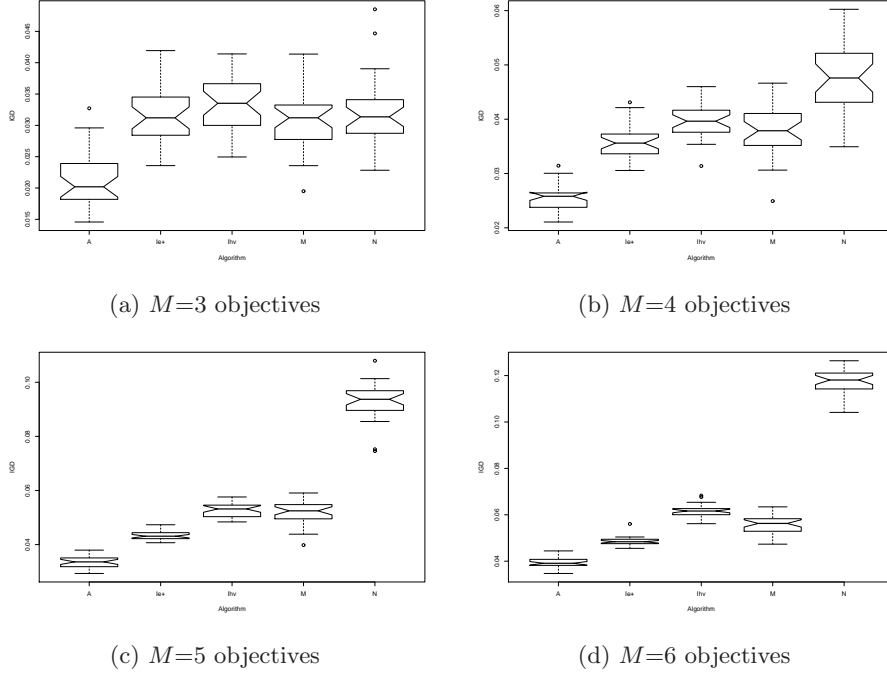


Fig. 5. IGD. Algorithms AεSεH (A), IBEA $I_{\varepsilon+}$ ($I_{\varepsilon+}$), IBEA I_{HD} (I_{HD}), NSGA-II (N) and MOEA/D (M).

Pareto optimal set, so we compute IGD taking as reference the non-dominated set obtained from the non-dominated solutions found by all algorithms.

First, looking at IGD in Fig.5, note that AεSεH achieves better (lower) IGD than the other algorithms in 3, 4, 5 and 6 objectives. In 3 objectives, IBEA $I_{\varepsilon+}$, IBEA I_{HD} , MOEA/D and NSGA-II achieve similar IGD . However, for $M > 3$ objectives IBEA $I_{\varepsilon+}$ is the second best algorithm in terms of IGD . For $M = 4$ and $M = 5$ there is not much difference between IBEA I_{HD} and MOEA/D. However, for $M = 6$ MOEA/D is significantly better than IBEA I_{HD} . NSGA-II is overall the worst algorithm

Next, looking at coverage C in Fig.6, note that for $M = 3$ $C(A, \cdot) > C(\cdot, A)$ for all algorithms Iε+, Ihd, M, and N. This indicates that solutions of AεSεH dominate more solutions of the other algorithms and fewer solutions of AεSεH are dominated by solutions of the other algorithms. Increasing the number of objectives above 3, the dominance gap between AεSεH and MOEA/D and between AεSεH and NSGA-II increase. However, fewer solutions by IBEA algorithms are dominated by AεSεH. For example in $M = 6$ objectives, in average around 3% of IBEA I_{HD} 's solutions are dominated by AεSεH and around 20% of AεSεH's solutions are dominated by IBEA I_{HD} . Between the two IBEA algorithms, C is slightly better for I_{HD} than for $I_{\varepsilon+}$. This however depends strongly on the value set for κ in IBEA.

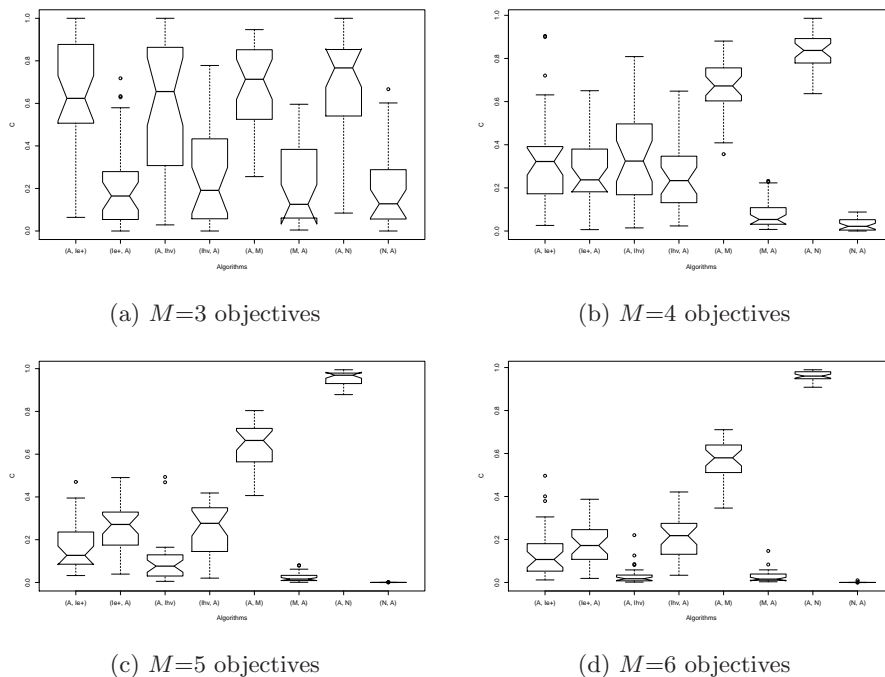


Fig. 6. C metric. Algorithms $A\epsilon S\epsilon H$ (A), $IBEA I_{\epsilon+}$ ($I_{\epsilon+}$), $IBEA I_{HD}$ (I_{HD}), NSGA-II (N) and MOEA/D (M).

These results complement our analysis of the previous section and confirms our expectation that too much exploration by MOEA/D could be detrimental to its performance in larger landscapes. It also shows that IBEA can find a subset of well converged solutions. However, it does it at the expense of not finding a well spread set of solutions. $A\epsilon S\epsilon H$ seems to have a good balance between convergence and diversity, which favors its scalability to larger landscapes. It will be interesting to find ways to control the exploration/exploitation features of the algorithms studied here to improve their performance, whether we scale up the objective space or the search space.

5 Conclusions

This work analyzed and compared the performance of MOEA/D, IBEA using the binary additive ϵ and the hypervolume difference indicators, and $A\epsilon S\epsilon H$ for many-objective optimization. We traced the dynamics of the algorithms in small MNK-landscapes, performed an off-line analysis of the Pareto optimal solutions discovered and dropped at each generation, and compared the algorithms for their ability to generate high-resolution approximations of the Pareto optimal set. Our analysis in small landscapes showed that exploration in MOEA/D is more intense than in $A\epsilon S\epsilon H$. This favors MOEA/D in small landscapes as we in-

crease the number of objectives, where it is relatively easy to hit the Pareto optimal set and exploration is more important to increase the resolution of the approximation. However, in large landscapes too much exploration hinders MOEA/D and $A\epsilon S\epsilon H$ generates approximations with better convergence and diversity, regardless of the number of objectives. IBEA converges to the central region of objective space, achieving low resolutions in small landscapes. In large landscapes this results in a subset of solutions with very good convergence properties, but poorly spread. In the future we would like to find ways to control the exploration/exploitation features of the algorithms to improve their performance when we scale up the objective and search spaces.

References

1. E. Hughes, “MSOPS-II: A General-Purpose Many-objective Optimiser”, *Proc. IEEE Congress on Evolutionary Computation (CEC 2007)*, pp. 3944-3951, 2007.
2. Q. Zhang and H. Li, MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition, *IEEE Trans. on Evol. Computation*, 11(6), 712–731, 2007.
3. E. Zitzler, S. Kunzli, “Indicator-Based Selection in Multiobjective Search”, *Proc. 8th Int’l Conference on Parallel Problem Solving from Nature PPSN VIII*, Springer, Lecture Notes in Computer Science, vol. 3242, pp. 832-842, 2004.
4. N. Beume, B. Naujoks, M. Emmerich, “SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume”, *European Journal on Operational Research*, vol. 181, no.3, pp.1653-1669, 2007.
5. C. Igel, N. Hansen, S. Roth, “Covariance Matrix Adaptation for Multi-objective Optimization”, *Evolutionary Computation*, 15(1), 1-28, 2007.
6. M. Laumanns, L. Thiele, K. Deb, E. Zitzler, “Combining Convergence and Diversity in Evolutionary Multi-objective Optimization”, *Evolutionary Computation*, 10(3), 263-282, 2002.
7. D. Hadka and P. Reed, “Borg: An Auto-adaptive Many-objective Evolutionary Computing Framework”, *Evolutionary Computation*, 2(2), 231-259, 2013.
8. H. Aguirre, A. Oyama, and K. Tanaka, “Adaptive ϵ -Sampling and ϵ -Hood for Evolutionary Many-Objective Optimization”, *Proc. 7th Int’l Conf. on Evolutionary Multi-Criterion Optimization*, Springer, LNCS, vol.7811, pp. 322-336, 2013.
9. K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, “A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II”, *KanGAL report 200001*, 2000.
10. H. Aguirre and K. Tanaka, “Insights on Properties of Multi-objective MNK-Landscapes”, *Proc. 2004 IEEE Congress on Evolutionary Computation*, IEEE Service Center, pp.196–203, 2004.
11. H. Aguirre, A. Liefooghe, S. Verel and K. Tanaka, “An Analysis on Selection for High-Resolution Approximations in Many-objective Optimization”, *Proc. 13th Int’l Conference on Parallel Problem Solving from Nature*, Springer, Lecture Notes in Computer Science, vol. 8672, pp. 487-497, 2014.
12. S. Zapotecas Martínez, H. Aguirre, K. Tanaka and C. Coello, “On the Low-Dyscrepancy Sequences and Their use in MOEA/D for High Dimensionality Objective Spaces”, *Proc. 2015 IEEE Congress on Evolutionary Computation*, IEEE Press, to appear, 2015.