An Analysis on Selection for High-Resolution Approximations in Many-Objective Optimization

Hernan Aguirre, Arnaud Liefooghe, Sébastien Verel, Kiyoshi Tanaka

▶ **To cite this version:**

Hernan Aguirre, Arnaud Liefooghe, Sébastien Verel, Kiyoshi Tanaka. An Analysis on Selection for High-Resolution Approximations in Many-Objective Optimization. Parallel Problem Solving from Nature - PPSN XIII, Sep 2014, Ljubljana, Slovenia. pp.487-497. hal-01066206

HAL Id: hal-01066206
https://hal.science/hal-01066206

Submitted on 24 Sep 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.

An Analysis on Selection for High-Resolution Approximations in Many-objective Optimization

Hernán Aguirre[1], Arnaud Liefooghe[2], Sébastien Verel[3], and Kiyoshi Tanaka[1]

[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 studies the behavior of three elitist multi- and many-objective evolutionary algorithms generating a high-resolution approximation of the Pareto optimal set. Several search-assessment indicators are defined to trace the dynamics of survival selection and measure the ability to simultaneously keep optimal solutions and discover new ones under different population sizes, set as a fraction of the size of the Pareto optimal set.

1 Introduction

In multi-objective optimization the aim of the optimizer is to find a good approximation of the Pareto optimal set (POS) in terms of convergence and diversity of solutions. Convergence dictates that solutions in the approximation must be either members of the POS or close to it in objective space. Diversity usually implies that solutions in the approximation should be evenly spaced in objective space, following the distribution of the POS.

In many-objective optimization, in addition to convergence and diversity, a third criterion also becomes a relevant aim of the optimizer. We call it the *resolution* of the approximation. The *resolution* is related to the number of points in the generated approximation of the POS. In many-objective problems the number of solutions in the POS increases exponentially [1] with the dimensionality of the objective space. In general, many more points are required to cover uniformly with the same density a higher dimensional space. However, the required resolution of the approximation could vary depending on the application domain and the task of the optimization within the problem solving approach. A low resolution of the approximation may suffice in some domain applications. For example, domains where the formulation of the problem is already well understood and a solution has to be found and implemented regularly, such as the daily operational schedule of machines and the jobs assigned to them in a manufacturing plant. In these domains, the optimizer is often required to provide alternative exact solutions and too many of them could overwhelm a decision

maker (operations manager) that must suggest a prompt course of action. In other application domains a high resolution of the approximation is required. For example in design optimization, where the problem-solving cycle often starts with multiple, sometimes ill defined, problem formulations and uses optimization as a tool to validate the understanding of the problem and to discover new features about it. In these application domains it is not unusual to require that the optimizer provides approximations of the POS with tens of thousands or even hundreds of thousands of solutions. These approximations are subjected to data mining and analysis to verify and improve the problem formulation itself, understand the tradeoffs between variables and objectives, and extract valuable design knowledge [2]. Thus, a many-objective optimizer should also aim to find an approximation with a *resolution* that properly captures the POS, with enough points to provide a useful description of it, depending on the dimensionality of the objective space and the optimization task at hand.

Many-objective optimization was initially attempted using evolutionary algorithms that proved effective for two and three objectives only to discover their lack of scalability. A significant part of the research effort has been understanding the reasons for their failure and improving them, particularly in terms of convergence. Recently, some many-objective optimizers are being proposed [3, 4]. However, the performance of the improved and newly proposed algorithms is commonly assessed using a relatively very small number of points focusing mostly on convergence and/or diversity. The resolution of the approximation in many-objective optimization has not been deeply studied and it is not clear the capabilities and behavior of the algorithms under this additional important criterion.

In this work we analyze the behavior of three elitist multi- and many-objective evolutionary algorithms generating a high-resolution approximation of the POS. We define a basic indicator for resolution, the accumulated gain of the population, and several generational search-assessment indices respect to the POS. We trace the dynamics of survival selection and study the ability to simultaneously keep Pareto optimal (PO) solutions in the population and find new ones to improve the resolution of the approximation, setting population size as a fraction of the size of the POS. We use MNK-landscapes with $3 - 6$ objectives and 20 bits, for which it is possible to know by enumeration all PO solutions.

2 Methodology

An important objective of this work is to analyze the ability of the algorithms to generate a high-resolution approximation of the POS. A simple and basic indicator for resolution is to count the number of PO solutions found by the algorithms. In many-objective problems is likely that the population size is considerable smaller than the size of the POS. Thus, to achieve a good resolution the algorithms should first be able to hit the POS with some members of the population and then continue discovering other PO solutions. The ability to converge towards the POS is a very important feature of the algorithm. In this

Table 1. Number of Pareto optimal solutions $|POS|$ and number of non-dominated *Fronts* in the landscapes with $M = 3$, 4, 5, and 6 objectives. Also, fraction of $|P|$ / $|POS|$ (in %) for various population sizes $|P|$ investigated in this study.

			$\|P\|$ / $\|POS\|$ (%)								
M	$\|POS\|$	$Fronts$	50	100	200	500	1,000	2,000	4,000	5,600	11,200
3	152	258	32.9	65.8	132.6						
4	1,554	76	3.2	6.4	12.9	32.2	64.4				
5	6,265	29	0.8	1.6	3.2			31.9	63.8		
6	16,845	22	0.3	0.6	1.2					33.2	66.5

study, we focus our attention mostly on the ability of the algorithm to continue discovering new PO solutions assuming that the algorithms can converge to the POS.

To evaluate this ability we use four MNK-landscapes [1] randomly generated with $M = 3$, 4, 5, 6 objectives, $N = 20$ bits, and $K = 1$ epistatic bit. In small landscapes with low non-linearity it is relatively simple for the algorithm to hit the optimal set. It is also possible to enumerate them and know the POS in order to analyze the dynamic of the algorithms respect to the optimum set. The exact number of PO solutions 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 here.

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 $\mathcal{F}_1(t)$, $t = 0, 1, \cdots, T$, making sure no duplicate solutions are included. In general, the approximation at generation t is given by

$$\mathcal{A}(t) = \{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{X}(t) = \mathcal{A}(t-1) \cup \mathcal{F}_1(t) \setminus \mathcal{A}(t-1) \cap \mathcal{F}_1(t) \wedge \nexists \boldsymbol{y} \in \mathcal{X}(t)\, \boldsymbol{y} \succeq \boldsymbol{x}\} \quad (1)$$

$$\mathcal{A}(0) = \mathcal{F}_1(0), \quad (2)$$

where $\boldsymbol{y} \succeq \boldsymbol{x}$ denotes solution $\boldsymbol{y}$ Pareto dominates solution $\boldsymbol{x}$. The basic resolution index α of the approximation at generation t is,

$$\alpha(t) = \frac{|\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{A}(t) \wedge \boldsymbol{x} \in POS\}|}{|POS|}, \quad (3)$$

which gives the fraction of the accumulated number of 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. Similarly, the accumulated population gain at generation t can be expressed as

$$\beta(t) = \frac{|\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{A}(t) \wedge \boldsymbol{x} \in POS\}|}{|P|}. \quad (4)$$

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	$\|\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{F}_1(t) \wedge \boldsymbol{x} \in POS\}\| \;/\; \|P\|$	PO solutions
τ_t^-	$\|\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{F}_1(t) \wedge \boldsymbol{x} \in \mathcal{F}_1(t-1) \wedge \boldsymbol{x} \in POS\}\| \;/\; \|P\|$	Old PO solutions
τ_t^+	$\|\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{F}_1(t) \wedge \boldsymbol{x} \notin \mathcal{F}_1(t-1) \wedge \boldsymbol{x} \in POS\}\| \;/\; \|P\|$	Possibly new PO solutions
τ_t^*	$\|\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{F}_1(t) \wedge \boldsymbol{x} \notin \cup_{k=1}^{t-1}\mathcal{F}_1(k) \wedge \boldsymbol{x} \in POS\}\| \;/\; \|P\|$	Absolutely new PO solutions
δ_t	$\|\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{F}_1(t-1) \wedge \boldsymbol{x} \notin \mathcal{F}_1(t) \wedge \boldsymbol{x} \in POS\}\| \;/\; \|P\|$	Dropped PO solutions
γ_t	$\|\{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{F}_1(t) \wedge \boldsymbol{x} \notin POS\}\| \;/\; \|P\|$	Non-dominated, not PO sol.

For our analysis on the dynamics of the algorithm, we compare the sets $\mathcal{F}_1(t)$ with the POS to determine which solutions in $\mathcal{F}_1(t)$ are Pareto optimal and compute several generational search-assessment indices I_t (τ_t, τ_t^*, τ_t^+, τ_t^-, δ_t, γ_t), as shown in Table 2. Note that these generational indexes are expressed as a fraction of the population size $|P|$. In this work we analyze them and their average value $\bar{I}$ ($\bar{\tau}$, $\bar{\tau}^*$, $\bar{\tau}^+$, $\bar{\tau}^-$, $\bar{\delta}$, $\bar{\gamma}$) taken over all generations computed as $\bar{I} = \frac{1}{T+1}\sum_{t=0}^{T} I_t$.

In this work we analyze NSGA-II [5], IBEA [6], and the Adaptive ε-Sampling and ε-Hood algorithm [4]. In the following we briefly describe these algorithms, particularly fitness assignment, survival selection, and parent selection.

3 Algorithms

3.1 NSGA-II

NSGA-II is an elitist algorithm that uses Pareto dominance and crowding estimation of solutions for survival and parent selections. To compute fitness of the individuals, the algorithm joins the current population $\mathcal{P}_t$ with its offspring $\mathcal{Q}_t$ and divide it in non-dominated fronts $\mathcal{F} = \{\mathcal{F}_i\}, i = 1, 2, \cdots, N_F$ using the non-dominated sorting procedure. It also calculate the crowding distance d_j of solutions within the fronts $\mathcal{F}_i$. The fitness of j-th solution in the i-th front is $\text{Fitness}(\boldsymbol{x}_j) = (i, d_j)$, where the front number is the primary rank and crowding distance the secondary rank. Survival selection is performed by copying iteratively the sets of solutions $\mathcal{F}_i$ to the new population $\mathcal{P}_{t+1}$ until it is filled. If the set $\mathcal{F}_i$, $i > 1$, overfills the new population $\mathcal{P}_{t+1}$, the required number of solutions are chosen based on their secondary rank d_j. Parent selection for reproduction consists of binary tournaments between randomly chosen individuals from $\mathcal{P}_{t+1}$ using their primary rank i to decide the winners, breaking ties with their secondary rank d_j.

3.2 IBEA (Indicator-Based Evolutionary Algorithm)

The main idea of IBEA [6] is 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 $\boldsymbol{x}$ is assigned a fitness value measuring the "loss in quality" if $\boldsymbol{x}$ was removed from the population P, i.e., $\text{Fitness}(\boldsymbol{x}) = \sum_{\boldsymbol{x}' \in P \setminus \{\boldsymbol{x}\}} (-e^{-I(\boldsymbol{x}', \boldsymbol{x})/\kappa})$, 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.

Different indicators can be used within IBEA. We here choose to use the binary additive ϵ-indicator ($I_{\epsilon+}$), as defined by the original authors [6].

$$I_{\epsilon+}(\boldsymbol{x}, \boldsymbol{x}') = \max_{i \in \{1,\ldots,n\}} \{f_i(\boldsymbol{x}) - f_i(\boldsymbol{x}')\} \tag{5}$$

$I_{\epsilon+}(\boldsymbol{x}, \boldsymbol{x}')$ gives the minimum value by which a solution $\boldsymbol{x} \in P$ has to, or can be translated in the objective space in order to weakly dominate another solution $\boldsymbol{x}' \in P$. More information about IBEA can be found in [6].

3.3 The AεSεH

Adaptive ε-Sampling and ε-Hood (AεSεH) [4] is an elitist evolutionary many-objective algorithm that applies ε-dominance principles for survival selection 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 $\mathcal{F} = \{\mathcal{F}_i\}, i = 1, 2, \cdots, N_F$ 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 $\mathcal{P}_{t+1}$.

After survival selection there is not an explicit ranking that could be used to bias mating. Rather, for parent selection the algorithm first uses a procedure called *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 than can be 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 will be selected randomly from the second neighborhood in the list, 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 that 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 AεSεH can be found in [4].

4 Experimental Results and Discussion

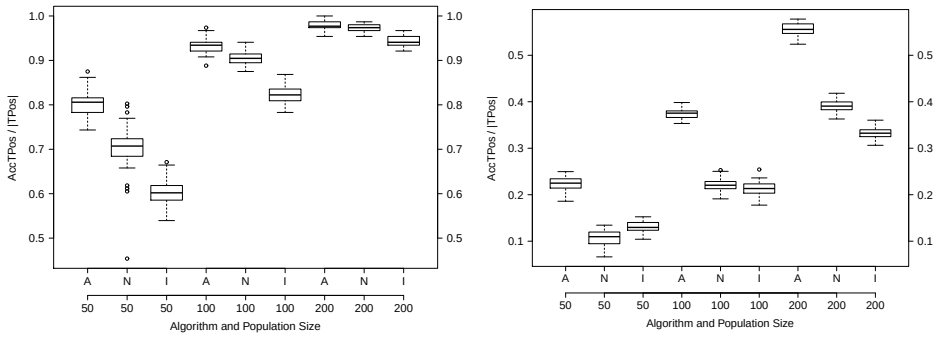
4.1 Operators of Variation and Parameters

In all algorithms we use two point crossover with rate $pc = 1.0$, and bit flip mutation with rate $pm = 1/n$. In AεSεH we set the reference neighborhood size H_{size}^{Ref} to 20 individuals. The mapping function $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}^{'}(\boldsymbol{x})$ used for ε-dominance in ε-sampling truncation and ε-hood creation is additive, $f_i^{'} = f_i + \varepsilon, i = 1, 2, \cdots, m$. For IBEA, the scaling factor is set to $\kappa = 0.001$. The algorithms run for $T = 100$ generations. Results analyzed here were obtained from 30 independent runs of the algorithms.

4.2 Accumulated number of Pareto optimal solutions found

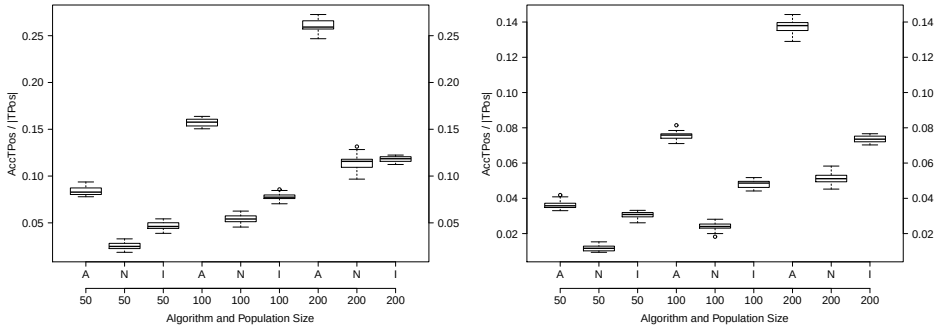
Fig.1 shows the the basic resolution index $\alpha(T)$ of the approximation at the end of the run, i.e. the ratio of accumulated number of PO solutions found to the size of the POS. Results are shown for 3, 4, 5, and 6 objectives using population sizes of $\{50, 100, 200\}$. Similarly, Fig.2 shows results for 5, and 6 objectives using larger populations sizes, between 500 and $11,2000$ individuals.

Note that AεSεH finds many more Pareto optimal solutions than NSGA-II and IBEA for all population sizes and number of objectives tried here, whereas NSGA-II finds more solutions than IBEA when population sizes are relatively a large fraction of the size of the POS. See Fig.1 (a) and Fig.2 (a)-(b) where population sizes correspond roughly to 33%, 66%, and 133% of the POS for 3 objectives and 33% and 66% for 5 and 6 objectives, as shown in *Table* 1. On the contrary, IBEA finds more solutions than NSGA-II when population sizes are relatively a small fraction of the POS. See Fig.1 (c)-(d) where population sizes $\{50, 100, 200\}$ are used in 5 and 6 objectives, which correspond to fractions in the range $0.3\% - 3.2\%$ of the POS. In 4 objectives, Fig.1 (b), an interesting transition can be observed. When the smallest population is used, i.e. 50 individuals $\sim 3.2\%$ of POS, IBEA finds more solutions than NSGA-II. For a population size of 100 $\sim 6.4\%$ of POS NSGA-II finds a slightly larger number of solutions than IBEA. For a population size of 200 $\sim 12.9\%$ of POS, NSGA-II finds a significant larger number of solutions than IBEA.



(a) M=3 objectives

(b) M=4 objectives



(c) M=5 objectives

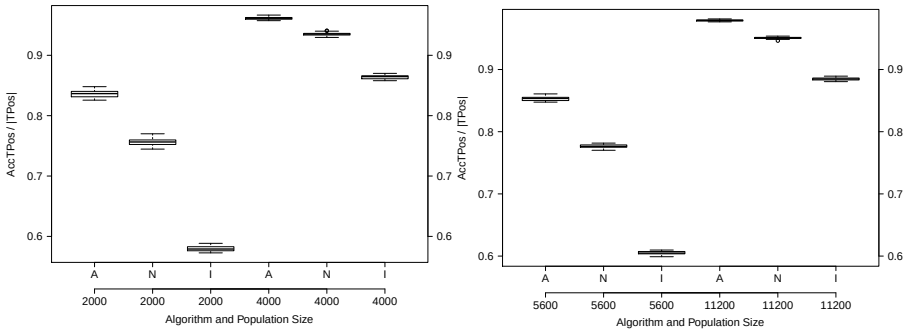
(d) M=6 objectives

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), NSGA-II (N) and IBEA (I).

The gap between AεSεH and the other two algorithms augments when the population size increases within a range in which it still is a small fraction of the POS, as shown in Fig.1 (b)-(d) where the ranges in which population size increase are $3.2\% - 12.9\%$, $0.8\% - 3.2\%$, and $0.3\% - 1.2\%$ of POS for 4, 5, and 6 objectives, respectively. On the other hand, the gap reduces when population size increases within a range in which it is a large fraction of the POS, as shown in Fig.1 (a) and Fig.2 (a)-(b) where the ranges in which population size increase are roughly $33\% - 133\%$ of POS for 3 objectives and $33\% - 66\%$ for 5 and 6 objectives.

4.3 Generational search assessment indices

Fig.3 (a)-(d) show boxplots of some $\bar{I}$ indexes computed from data obtained in 30 independent runs of the algorithms iterating $T = 100$ generations. Results



(a) M=5 objectives (b) M=6 objectives

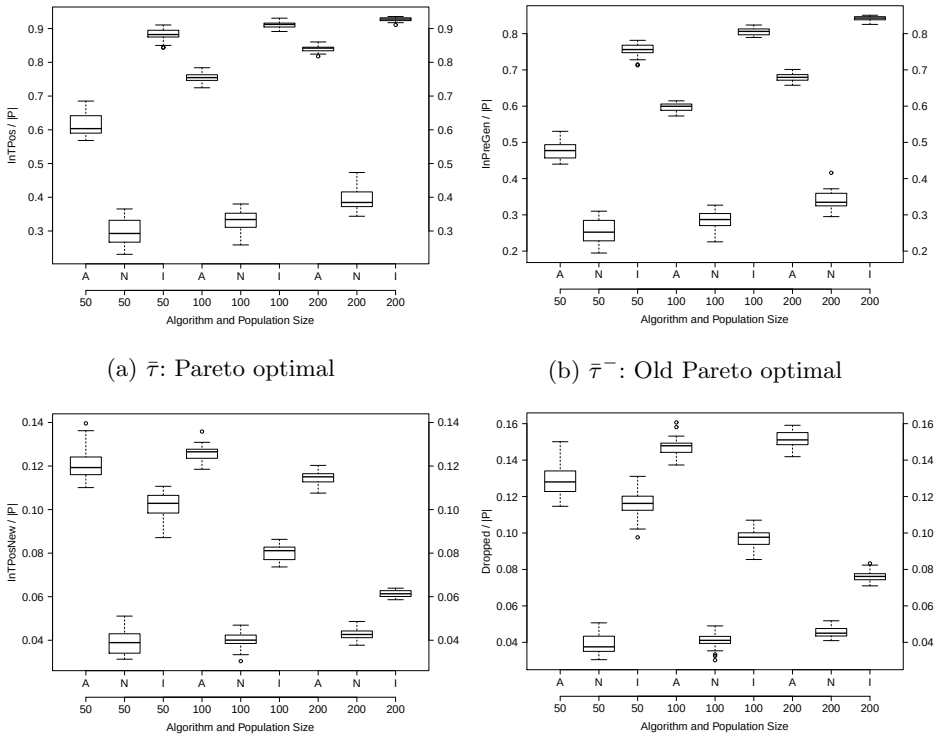
Fig. 2. Ratio of accumulated number of Pareto optimal solutions found to the size of the POS. Population sizes $\{500, 1000\}$, $\{2000, 4000\}$, and $\{5600, 11200\}$ for 4, 5, and 6 objectives, respectively. Algorithms AεSεH (A), NSGA-II (N) and IBEA (I).

are shown for 6 objectives landscapes using population sizes {50, 100, 200}, which are relatively small compared to the POS. From these figures important observations are as follow.

In average, at each generation, IBEA contains in its population a very large number of PO solutions compared to AεSεH and NSGA-II as shown in Fig.3 (a). Note that the median of index $\bar{\tau}$ for IBEA is in the range 0.88-0.95, whereas the ranges for AεSεH and NSGA-II are between 0.6-0.85 and 03-0.4, respectively.

However, the number of old PO solutions (PO solutions present in the current population and also in the population of the previous generation) for IBEA is much larger than for AεSεH and NSGA-II, as shown in Fig.3 (b). Note that the median of $\bar{\tau}^-$ is in the range 0.75-0.85 for IBEA, whereas $\bar{\tau}^-$ is in the range 0.48-0.68 for AεSεH and 0.25-0.35 for NSGA-II.

In fact, the generational average number of absolutely new PO solutions (PO solutions in the current population that have not been discovered in previous generations) is larger for AεSεH than for IBEA and NSGA-II, as shown in Fig.3 (c). Note that the median of index $\bar{\tau}^*$ for AεSεH is around 0.12, whereas for IBEA it reduces with population size from 0.10 to 0.06 and slightly increases for NSGA-II from 0.039 to 0.04. The similar $\bar{\tau}^*$ values by AeSEH are a good sign of robustness to population size variations, i.e. a similar discovery rate could be expected with various population sizes. On the contrary, IBEA's discovery rate could reduce significantly with population size. If the evaluation of the algorithms is done based only on the points included in the population at a given generation, as it is often the case, IBEA is likely to contain more PO solutions than AεSεH, as shown in Fig.3 (a), and therefore be considered a better algorithm. However, AεSεH finds twice as many PO solutions than IBEA, as shown in Fig.1 (d). These results show that IBEA could be a good algorithm for finding a low resolution approximation of the POS, but for high resolutions is not efficient. In



(a) $\bar{\tau}$: Pareto optimal

(b) $\bar{\tau}^{-}$: Old Pareto optimal

(c) $\bar{\tau}^{*}$: Absolutely new Pareto optimal

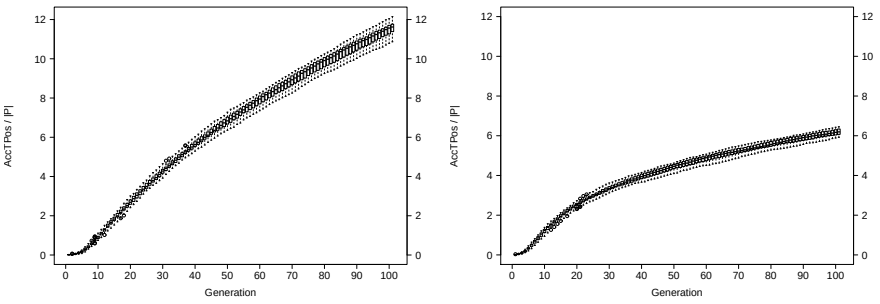
(d) $\bar{\delta}$: Dropped Pareto optimal

Fig. 3. Boxplots of average generational search-assessment indices in 30 runs. Population sizes $\{50, 100, 200\}$, 6 objectives, $T = 100$ generations. Algorithms AεSεH (A), NSGA-II (N) and IBEA (I).

general, these results show the importance of properly evaluating the algorithms according to the aim of the optimization task at hand.

The index of dropped PO solutions $\bar{\delta}$ (PO solutions present in the population of the previous generation that are not included in the current population after truncation selection) shows a trend vey similar to the one observed for the index $\bar{\tau}^{*}$, as shown in Fig.3 (d). Dropping superior solutions in favor of solutions that appear non-dominated in the population but are inferior in the landscape could be seen as a selection weakness. However, this could also be a source of exploration. This deserves further research.

The accumulated population gains $\beta(t)$ for AεSεH and IBEA are illustrated in Fig.4 for population size 200 and 6 objectives. Note that just after 20 generation the gain by AεSεH is already larger than by IBEA. At the end of the run, AεSεH is able to generate an approximation twelve times the size of its population, whereas IBEA is able to generate an approximation 6 times the size of its population.



(a) AeSeH (b) IBEA

Fig. 4. Accumulated population gains $\beta(t)$ over the generations. Population size 200, 6 objectives. Algorithms AεSεH and IBEA.

5 Conclusions

This work has studies the behavior of NSGA-II, IBEA and AεSεH generating a high-resolution approximation of the POS. The study has clarified the ability and efficiency of the algorithms assuming scenarios where it is relatively easy to hit the POS, showing the importance to properly assess algorithm's performance according to the task of the optimizer in many objective optimization. In the future, we would like to extend our study to larger landscapes in order to understand the behavior of selection in scenarios where the convergence ability towards the POS is determinant to achieve a good resolution. Also, we would like to study other indicators for IBEA and other many-objective algorithms.

References

1. 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.
2. Y. Nishio, A. Oyama, Y. Akimoto, H. Aguirre and K. Tanaka. "Many-Objective Optimization of Trajectory Design for DESTINY Mission", Learning and Intelligent Optimization Conference, Lecture Notes in Computer Science, Springer, 2014.
3. D. Hadka and P. Reed, "Borg: An Auto-adaptive Many-objective Evolutionary Computing Framework", Evol. Computation, vol.2, no. 2, pp.231-259, 2013.
4. 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 (EMO 2013), Lecture Notes in Computer Science, Springer, vol.7811, pp. 322-336, 2013.
5. 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.
6. E. Zitzler, S. Kunzli, "Indicator-based Selection in Multiobjective Search", Proc. 8th Int'l Conference on Parallel Problem Solving from Nature - PPSN VIII, Lecture Notes in Computer Science, Springer, vol. 3242, pp. 832-842, 2004.