Adaptive ε-Sampling and ε-Hood for Evolutionary Many-Objective Optimization

Hernán Aguirre[1], Akira Oyama[2], and Kiyoshi Tanaka[1]

[1] Faculty of Engineering, Shinshu University
4-17-1 Wakasato, Nagano, 380-8553 Japan
[2] Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency
{ahernan,ktanaka}@shinshu-u.ac.jp, oyama@flab.isas.jaxa.jp

Abstract. Many-objective problems are becoming common in several real-world application domains and there is a growing interest to develop evolutionary many-objective optimizers that can solve them effectively. Studies on selection for many-objective optimization and most recently studies on the characteristics of many-objective landscapes, the effectiveness of operators of variation, and the effects of large populations have proved successful to advance our understanding of evolutionary many-objective optimization. This work proposes an evolutionary many-objective optimization algorithm that uses adaptive ε-dominance principles to select survivors and also to create neighborhoods to bias mating, so that solutions will recombine with other solutions located close by in objective space. We investigate the performance of the proposed algorithm on DTLZ continuous problems, using a short number of generations to evolve the population, varying population size from 100 to 20000 individuals. Results show that the application of adaptive ϵ-dominance principles for survival selection as well as for mating selection improves considerably the performance of the optimizer.

1 Introduction

Many-objective problems are becoming common in several real-world application domains and there is a growing interest to develop evolutionary many-objective optimizers (EMyOs) that can solve them effectively. One such application domains is multi-objective design exploration for real-world design optimization. Here, a large number of Pareto optimal solutions that give a good representation of the true Pareto front in terms of convergence, spread, and distribution of solutions along the front are essential to extract relevant knowledge about the problem. This knowledge, rather than a particular precise solution, is valuable to establish trade-offs and hotspots regions for objectives and design variables in order to provide useful guidelines to designers during the selection of alternative designs, and to facilitate the implementation of the finally chosen design. In these applications it is also common that the evaluation of solutions is computationally expensive and takes a long time to calculate it, which prohibits running the evolutionary algorithm for a large number of generations. Thus, in real-world

R.C. Purshouse et al. (Eds.): EMO 2013, LNCS 7811, pp. 322–336, 2013.

applications of many-objective optimization we have to face the difficulties imposed by the increased complexities of large-dimensional spaces and are often constrained by time.

It is well known that conventional evolutionary multi-objective optimizers (EMOs) [1, 2] scale up poorly with the number of objectives of the problem, which is often attributed to the large number of non-dominated solutions and the lack of effective selection and diversity estimation operators to discriminate appropriately among them, particularly in dominance-based algorithms. Selection, indeed, is a fundamental part of the algorithm and has been the subject of several studies, leading to improve the performance of conventional EMOs on many-objective problems. However, finding trade-off solutions that satisfy simultaneously the three properties of convergence, spread, and distribution is especially difficult on many-objective problems. In fact, most EMOs with improved selection strategies for many-objective optimization proposed recently compromise one in favor of other [3]. In addition to selection, recent studies on the characteristics of many-objective landscapes, the effectiveness of operators of variation, and the effects of large populations [4–7] have proved successful to advance our understanding of evolutionary many-objective optimization.

In previous work, we have studied the behavior of Adaptive ε-Box with Neighborhood Recombination [6], an algorithm built under the framework of NSGA-II [8]. The algorithm uses ε-box non-dominated sorting with a logarithmic function that maps solutions to a grid [9] and selects for survival just one solution per non-dominated ε-box. It also keeps a list of neighbors for each solution to restrict mating. This algorithm improves considerably the performance of NSGA-II. However, the logarithmic function originally proposed in [9] introduces a too strong bias towards the edges of objective space. In addition, though nearby individuals recombine, in this approach the search effort is not evenly balanced towards all regions of objective space. Moreover, it uses crowding distance as secondary ranking, which does not scale up well in high dimensional spaces.

In this work, we propose an EMyO that uses adaptive ε-dominance principles to select survivors and also to create neighborhoods to bias mating for recombination. The method used for survival selection is based on ε-sampling that selects solutions randomly from the set of non-dominated solutions, eliminating solutions that are ε-dominated by the sampled solutions. The motivation to use ε-sampling is that surviving solutions are spaced following the distribution implicit in the mapping function used for ε-dominance and the search effort could be balanced according to such distribution. In this work we use an additive mapping function that induces a uniform distribution of solutions, aiming to cover all regions of objective space. The method to create neighborhoods is also based on ε-dominance. Here, a randomly sampled solution from the surviving population and its ε-dominated solutions determine the neighborhood, so that recombination can take place between individuals located nearby in objective space. The motivation to restrict mating is to enhance the effectiveness of recombination in many-objective problems, where the difference in variable space between individuals in the population is expected to be larger than in multi-objective problems and therefore more disruptive for recombination. In addition, the method gives more reproductive opportunities to individuals located in under-represented regions to balance the search effort towards all regions.

We investigate the performance of the proposed algorithm on DTLZ continuous problems, varying population size from very small to large -from 100 to 20000 individuals. We assume scenarios in which computational time to calculate fitness could be very large and thus use a very short number of generations to evolve the population. We also assume that all individuals in the population can be evaluated simultaneously in parallel. As a reference for comparison, we include results by Adaptive ε-Box with Neighborhood Recombination showing that the proposed algorithm performs significantly better in terms of convergence and distribution of solutions.

2 Proposed Method

2.1 Concept

In many-objective landscapes the number of solutions in the Pareto optimal set increases almost exponentially [10, 11] with the number of objectives. Keeping fixed the size of the variable space, an increase in the number of objectives also implies that these large number of Pareto optimal solutions become spread over broader regions in variable space [4]. Analysis of many-objective landscapes show that this is the case not only for the global Pareto set containing optimal solutions, but also for local Pareto sets containing suboptimal solutions [10, 11]. These characteristics of many-objective landscapes are reflected on the dynamics of the optimizer and are directly correlated to the effectiveness of the operators of selection and variation. For example, the large number of non-dominated solutions makes dominance-based selection random and their spread on variable space causes recombination to recombine distant solutions making it too disruptive and ineffective. Hence, the characteristics of many-objective landscapes must be carefully considered when we design our algorithms.

The proposed method aims to perform an effective search on many-objective landscapes by using dominance and ε-sampling for survival selection to get a well distributed subset of non-dominated solutions, and ε-hood creation and ε-mating for parent selection to enhance the effectiveness of recombination. In multi-objective optimization, dominance is used for survival selection and to rank solutions in the elite surviving population, so that mating selection can give more reproductive opportunities to dominant individuals. However, this is impractical in many-objective optimization. In the proposed method, dominance acts as a mean to eliminate dominated solutions during survival selection and leave the non-dominated ones for further processing, but it has no role ranking the surviving population. For most part of the evolution, survival selection is achieved by ε-sampling, which samples randomly from the large set of non-dominated solutions and eliminates solutions ε-dominated by the samples. The aim is to get a set of surviving solutions spaced according to the distribution implicit in the mapping function $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}^{'}(\boldsymbol{x})$ used for ε-dominance. Only during the few initial generations, where the number of non-dominated solutions is smaller than the size of the surviving population, ε-sampling plays no role during survival.

After survival selection, in the proposed method there is not an explicit ranking that could be used to bias mating. Rather, we use a procedure called ε-hood creation to cluster solutions in objective space. This method is also based on ε-dominance and is adaptive too. Here, a randomly sampled solution from the surviving population and its ε-dominated solutions determine the neighborhood, so that recombination can take place between individuals located nearby in objective space. The motivation to restrict mating is to enhance the effectiveness of recombination in many-objective problems, where the difference in variable space between individuals in the population is expected to be larger than in multi-objective problems and therefore more disruptive for recombination. In addition, to balance the search effort towards all regions, individuals located in under-represented regions are given more reproductive opportunities.

Summarizing, dominance acts as a mean to eliminate inferior solutions, ε-sampling gets a set of well distributed solutions from the large set of non-dominated solutions so that search effort could be uniformly distributed, ε-hood creation clusters elite solutions in objective space, ε-mating pairs nearby solutions to enhance the effectiveness of recombination, and reproduction gives more reproductive opportunities to individuals in under-represented regions.

2.2 Adaptive ε-Sampling and ε-Hood Evolutionary Many-Objective Optimizer (A$\varepsilon_s\varepsilon_h$EMyO)

In this section we explain the general flow of the proposed algorithm A$\boldsymbol{\varepsilon_s\varepsilon_h}$EMyO illustrated in **Procedure 1** and in the next sections we explain in detail its distinctive features. The proposed method uses $\boldsymbol{\varepsilon}$-dominance principles to truncate the population by sampling from the set of non-dominated solutions and also to create neighborhoods to bias mating for recombination, using parameters $\boldsymbol{\varepsilon_s}$ and $\boldsymbol{\varepsilon_h}$, respectively. The $\boldsymbol{\varepsilon_s}$ parameter for sampling during truncation is dynamically adapted to keep the number of sampled solutions $\boldsymbol{N_S}$ close to the population size $\boldsymbol{P_{size}}$. Similarly, the $\boldsymbol{\varepsilon_h}$ parameter for neighborhood creation is dynamically adapted to keep the number of neighborhoods $\boldsymbol{N_H}$ close to a user specified number $\boldsymbol{N_H^{Ref}}$. In addition to $\boldsymbol{\varepsilon_s}$ and $\boldsymbol{\varepsilon_h}$, their steps of adaptation are also adapted to properly follow the dynamics of the search. Thus, before its main loop, the algorithm sets the reference number of neighborhoods $\boldsymbol{N_H^{Ref}}$, initial values for $\boldsymbol{\varepsilon_s}$ and its step of adaptation $\boldsymbol{\Delta_s}$, and initial values for $\boldsymbol{\varepsilon_h}$ and its step of adaptation $\boldsymbol{\Delta_h}$. Next, it creates randomly the initial population. Then it iterates the main evolutionary loop.

The main loop starts by evaluating the offspring population $\boldsymbol{\mathcal{Q}}$. After offspring $\boldsymbol{\mathcal{Q}}$ is evaluated, non-dominated sorting is performed on the population that results from joining the current population $\boldsymbol{\mathcal{P}}$ and its offspring $\boldsymbol{\mathcal{Q}}$. The population of size $2\boldsymbol{P_{size}}$ sorted in non-dominated fronts $\boldsymbol{\mathcal{F}}$ is then truncated to obtain the surviving population $\boldsymbol{\mathcal{P}}$ of size $\boldsymbol{P_{size}}$ using a $\boldsymbol{\varepsilon}$-sampling truncation procedure set with parameter $\boldsymbol{\varepsilon_s}$.

Procedure 1. A$\varepsilon_s\varepsilon_h$EMyO

```
Require: Population size $P_{size}$, reference neighborhood size $H^{Ref}_{size}$
Ensure: $\mathcal{F}_1$, set of Pareto non-dominated solutions
$N^{Ref}_H \leftarrow P_{size}/H^{Ref}_{size}$   // set reference number of neighborhoods
$\varepsilon_s \leftarrow 0$, $\Delta_s \leftarrow \Delta_0$   // set $\varepsilon_s$-dominance factor and its step of adaptation
$\varepsilon_h \leftarrow 0$, $\Delta_h \leftarrow \Delta_0$   // set $\varepsilon_h$-dominance factor and its step of adaptation
$\mathcal{P} \leftarrow \emptyset$, $\mathcal{Q} \leftarrow random$   // initial populations $\mathcal{P}$ and $\mathcal{Q}$, $|\mathcal{Q}| = P_{size}$
repeat
  evaluation( $\mathcal{Q}$ )
  $\mathcal{F} \leftarrow$ non-dominated sorting( $\mathcal{P} \cup \mathcal{Q}$ )   // $\mathcal{F} = \{\mathcal{F}_i\},\ i = 1, 2, \cdots, N_F$
  $\{\mathcal{P}, N_S\} \leftarrow$ $\epsilon$-sampling truncation( $\mathcal{F}$, $\epsilon_s$, $P_{size}$ )   // $|\mathcal{P}| = |\mathcal{Q}| = P_{size}$
  $\{\epsilon_s, \Delta_s\} \leftarrow$ adapt ( $\epsilon_s$, $\Delta_s$, $P_{size}$, $N_S$ )
  $\{\mathcal{H}, N_H\} \leftarrow$ $\epsilon$-hood creation ( $\mathcal{P}$, $\epsilon_h$ ) // $\mathcal{H} = \{\mathcal{H}_j\},\ j = 1, 2, \cdots, N_H$
  $\{\epsilon_h, \Delta_h\} \leftarrow$ adapt ( $\epsilon_h$, $\Delta_h$, $N^{Ref}_H$, $N_H$ )
  $\mathcal{P}' \leftarrow$ $\epsilon$-hood mating( $\mathcal{H}$, $Psize$ )
  $\mathcal{Q} \leftarrow$ recombination and mutation( $\mathcal{P}'$ )
until termination criterion is met
return $\mathcal{F}_1$
```

The number of sampled solutions N_S and the population size P_{size} are used as reference to adapt ε_s and its step of adaptation Δ_s. Next, neighborhoods are created from the surviving population using a ε-hood creation procedure set with parameter ε_h. Similar to ε_s, ε_h and its step of adaptation Δ_h are adapted so that the number of created neighborhoods N_H would be close to a user specified reference number N^{Ref}_H. After the neighborhoods have been created, ε-hood mating creates a pool of mates $\mathcal{P}'$ by selecting solutions within the neighborhoods, so that a solution would recombine only with a solution that is close by in objective space. Next, the already defined mates are recombined and mutated to create the offspring population $\mathcal{Q}$ and the algorithm continues with the next generation until a termination criterion has been met.

2.3 ε-Sampling Truncation

Survival selection is implemented by the ε-sampling truncation method, illustrated in **Procedure 2**. This method receives the sets of solutions $\mathcal{F}$ created by non-dominated sorting and selects exactly P_{size} surviving solutions from them. In case the number of non-dominated solutions $|\mathcal{F}_1| > P_{size}$, it calls ε-sampling with parameter ε_s to get from $\mathcal{F}_1$ its extreme solutions $\mathcal{E}$, a subset of randomly sampled solutions $\mathcal{S}$ and their ε_s-dominated solutions $\mathcal{D}^{\varepsilon_s}$, as illustrated in **Procedure 3**.

The surviving population $\mathcal{P}$ always includes extreme solutions $\mathcal{E}$ and it is complemented with solutions from $\mathcal{S}$ and possibly from $\mathcal{D}^{\varepsilon_s}$. If $\mathcal{S}$ overfills $\mathcal{P}$, solutions in $\mathcal{S}$ are randomly eliminated as survivors. Otherwise, if after adding

$\mathcal{S}$ to $\mathcal{P}$ there is still room for some solutions, the required number are randomly chosen from $\mathcal{D}^{\epsilon_s}$. On the other hand, is $|\mathcal{F}_1| < P_{size}$, while there is room in $\mathcal{P}$ the sets of solutions $\mathcal{F}_i$ are copied iteratively to $\mathcal{P}$. The remaining solutions are chosen randomly from the set that did not fit completely in $\mathcal{P}$.

Procedure 2. ϵ-sampling truncation ($\mathcal{F}$, ϵ_s, P_{size})

Require: sets of non-dominated solutions $\mathcal{F} = \{\mathcal{F}_i\}, i = 1, 2, \cdots, N_F$, ϵ-dominance parameter ϵ_s and desired population size after truncation P_{size}

Ensure: Truncated population $\mathcal{P}$ obtained from $\mathcal{F}$ and number of sampled solutions including the extremes N_S

```
$\mathcal{P} \leftarrow \emptyset, N_S \leftarrow 0$
if $|\mathcal{F}_1| > P_{size}$ then
    $\{\mathcal{E}, \mathcal{S}, \mathcal{D}^{\epsilon_s}\} \leftarrow$ $\epsilon$-sampling( $\mathcal{F}_1$, $\epsilon_s$ )
    $N_S \leftarrow |\mathcal{E}| + |\mathcal{S}|$
    if $N_S > P_{size}$ then
        $\mathcal{X} \leftarrow \{x_r \in \mathcal{S} \mid r = rand(\ 1,\ |\mathcal{S}|\ ),\ |\mathcal{X}| = N_S - P_{size}\}$
        $\mathcal{P} \leftarrow \mathcal{E} \cup \mathcal{S} \setminus \mathcal{X}$
    else
        $\mathcal{X} \leftarrow \{x_r \in \mathcal{D}^{\epsilon_s} \mid r = rand(\ 1,\ |\mathcal{D}^{\epsilon_s}|\ ),\ |\mathcal{X}| = P_{size} - N_S\}$
        $\mathcal{P} \leftarrow \mathcal{E} \cup \mathcal{S} \cup \mathcal{X}$
    end if
else
    $\mathcal{P} \leftarrow \bigcup_{i=1}^{k} \mathcal{F}_i, \quad \sum |\mathcal{F}_i| < P_{size}$
    $\mathcal{X} \leftarrow \{x_r \in \mathcal{F}_{k+1} \mid r = rand(\ 1,\ |\mathcal{F}_{k+1}|\ ),\ |\mathcal{X}| = P_{size} - \sum_{i=1}^{k} |\mathcal{F}_i|\}$
    $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{X}$
end if
return $\mathcal{P}$ and $N_S$
```

Procedure 3. ϵ-sampling ($\mathcal{F}_1$, ϵ_s)

Require: Non-dominated solutions $\mathcal{F}_1$, ϵ-dominance parameter ϵ_s

Ensure: $\mathcal{E}$, $\mathcal{S}$ and $\mathcal{D}^{\epsilon_s}$, $\mathcal{E} \cup \mathcal{S} \cup \mathcal{D}^{\epsilon} = \mathcal{F}_1$. $\mathcal{E}$ and $\mathcal{S}$ contain extreme solutions and a randomly chosen sample of solutions from $\mathcal{F}_1$, respectively, whereas $\mathcal{D}^{\epsilon_s}$ contains solutions ϵ_s-dominated by those in $\mathcal{S}$. Maximization in all objectives is assumed

```
$\mathcal{E} \leftarrow \{x \in \mathcal{F}_1 \mid f_m(x) = \max(f_m(\cdot)), m = 1, 2, \cdots, M\}$ // extremes
$\mathcal{F}_1 \leftarrow \mathcal{F}_1 \setminus \mathcal{E}$
$\mathcal{D}^{\epsilon_s} \leftarrow \emptyset$
while $\mathcal{F}_1 \neq \emptyset$ do
    $z \leftarrow x_r \in \mathcal{F}_1 \mid r = rand(\ 1,\ |\mathcal{F}_1|\ )$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{z\}$    // add randomly chosen solution $z$ to sample
    $\mathcal{Y} \leftarrow \{y \in \mathcal{F}_1 \mid z \succeq^{\epsilon_s} y, z \neq y\}$    // solutions $\epsilon_s$-dominated by $z$
    $\mathcal{D}^{\epsilon_s} \leftarrow \mathcal{D}^{\epsilon_s} \cup \mathcal{Y}$
    $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \setminus \{\{z\} \cup \mathcal{Y}\}$
end while
return $\mathcal{E}$, $\mathcal{S}$, $\mathcal{D}^{\epsilon_s}$
```

2.4 ε-Hood Creation and ε-Hood Mating

Neighborhoods are created from the surviving population by the ε-hood creation procedure, which is also based on ε-dominance as illustrated in **Procedure 4**. This procedure randomly selects an individual from the surviving population and applies ε-dominance with parameter ε_h. A neighborhood is formed by the sampled solutions and its ε_h-dominated solutions. Neighborhood creation is repeated until all solutions in the surviving population have been assigned to a neighborhood.

Procedure 4. ε-hood creation ($\mathcal{P}$, ε_h)

Require: Population $\mathcal{P}$, ε-dominance parameter ϵ_h for neighborhood creation
Ensure: Neighborhoods $\mathcal{H} = \{\mathcal{H}_i\}$, $i = 1, 2, \cdots, N_H$

```
$\mathcal{H} \leftarrow \emptyset$
$i \leftarrow 0$
while $\mathcal{P} \neq \emptyset$ do
    $z \leftarrow x_r \in \mathcal{P} \mid r = rand( 1, |\mathcal{P}| )$   // $z$, a randomly chosen solution
    $\mathcal{Y} \leftarrow \{y \in \mathcal{P} \mid z \succeq^{\varepsilon_h} y, z \neq y\}$   // solutions $\varepsilon_h$-dominated by $z$
    $i \leftarrow i + 1$
    $\mathcal{H}_i \leftarrow \{\{z\} \cup \mathcal{Y}\}$   // $z$ and its $\varepsilon_h$-dominated solutions form the hood
    $\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{H}_i$
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{H}_i$
end while
$N_H \leftarrow i$
return $\mathcal{H}$, $N_H$
```

Procedure 5. ε-hood mating ($\mathcal{H}$, P_{size})

Require: Neighborhoods $\mathcal{H} = \{\mathcal{H}_i\}$, $i = 1, 2, \cdots, N_H$, and population size P_{size}
Ensure: Pool of mated parents $\mathcal{P}'$, $|\mathcal{P}'| = 2P_{size}$

```
$\mathcal{P}' \leftarrow \emptyset$
$i \leftarrow 1$
$j \leftarrow 0$
while $j < P_{size}$ do
    $\{y, z\} \leftarrow \{x_{r_1}, x_{r_2} \in \mathcal{H}_i \mid r_1 \wedge r_2 = rand( 1, |\mathcal{H}_i| ), r_1 \neq r_2\}$
    $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{y, z\}$
    $i \leftarrow 1 + (i \bmod N_H)$
    $j \leftarrow j + 1$
end while
return $\mathcal{P}'$
```

Mating for recombination is implemented by the procedure ε-hood mating illustrated in **Procedure 5**. Neighborhoods are considered to be elements of a list. To select two mates, first a neighborhood from the list is specified deterministically in a round-robin schedule. Then, two individuals are select randomly within the specified neighborhood, so that an individual will recombine with other individual that is located close by in objective space. Due to the round-robin schedule, the next two mates will be selected from the next neighborhood in the list. When the end of the neighborhood lists is reached, mating 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 scheduling individuals belonging to neighborhoods with fewer members have more recombination opportunities that those belonging to neighborhoods with more members. Once the pool of all mates $\mathcal{P}'$ has been established, they are recombined and mutated according to the order they were selected during mating.

2.5 Additive Epsilon Mapping $f(x) \mapsto^{\epsilon} f'(x)$

In this work we use an evenly spaced Additive mapping function $f(x) \mapsto^{\epsilon} f'(x)$ [5] for both ε-sampling and ε-hood creation. The Additive function maps $f(x)$ to $f'(x)$ by adding the same value ϵ to all coordinates f_i, independently of the position of $f(x)$ in objective space. This mapping in ε-sampling induces a distribution of solutions evenly spaced by ϵ. The expression for Additive mapping is as follows

$$f_i'(x) = f_i(x) + \epsilon, \quad i = 1, \cdots, m \tag{1}$$

2.6 Adaptation

The number of sampled solutions N_S by ε-sampling depends on the value set to ε_s (≥ 0). Larger values of ϵ_s imply that sampled solutions ε_s-dominate larger areas, increasing the likelihood of having more ε_s-dominated solutions excluded from the sample. The proposed algorithm adapts ε_s at each generation so that N_S is close to the population size P_{size}. The closer N_S is to P_{size}, the larger the number of surviving solutions that will be spaced according to the distribution implicit in the mapped function used for ε-dominance.

Similarly, the number of created neighborhoods N_H depends on the value set to ε_h (≥ 0). Larger values of ε_h imply that sampled solutions ε_h-dominate larger areas, increasing the likelihood of having more ε_h-dominated solutions that form its neighborhood, and therefore less created neighborhoods. The proposed algorithm adapts ε_h at each generation so that N_H is close to a user specified number N_H^{Ref}.

The adaptation rule, similar for both processes, is as follows. If $N > Ref$ it increases the step of adaptation $\Delta \leftarrow \min(\Delta \times 2, \Delta_{max})$ and $\epsilon \leftarrow \epsilon + \Delta$. Otherwise, if $N < Ref$ it decreases $\Delta \leftarrow \max(\Delta \times 0.5, \Delta_{min})$ and $\epsilon \leftarrow \max(\epsilon - \Delta, 0.0)$. In this work we set initial values $\epsilon_0 = 0.0$ and $\Delta_0 = 0.005$. Also, $\Delta_{max} = 0.05$ and $\Delta_{min} = 0.0001$.

In the case of adapting the parameter ε_s used for truncation, the above rule is called with $\varepsilon = \varepsilon_s$, $\Delta = \Delta_s$, $N = N_S$, and $Ref = P_{size}$. On the other hand, in the case of the parameter ε_h used for neighborhood creation, the above rule is called with $\varepsilon = \varepsilon_h$, $\Delta = \Delta_h$, $N = N_H$, and $Ref = N_H^{Ref}$.

3 Test Problems, Performance Indicators, and Experimental Setup

We study the performance of the algorithms in continuous functions DTLZ2, DTLZ3, and DTLZ4 of the DTLZ test functions family [12]. These functions are scalable in the number of objectives and variables and thus allow for a many-objective study. In our experiments, we vary the number of objectives from $m = 4$ to 6 and set the total number of variables to $n = (m - 1) + 10$. DTLZ2 has a non-convex Pareto-optimal surface that lies inside the first quadrant of the unit hyper-sphere. DTLZ3 and DTLZ4 are variations of DTLZ2. DTLZ3 introduces a large number of local Pareto-optimal fronts in order to test the convergence ability of the algorithm. DTLZ4 introduces biases on the density of solutions to some of the objective-space planes in order to test the ability of the algorithms to maintain a good distribution of solutions. For a detailed description of these problems the reader is referred to [12].

To evaluate the Pareto optimal solutions obtained by the algorithms we use the Generational Distance (GD) [13], which measures the convergence of solutions to the true Pareto front using equation 2, where P denotes the set of Pareto optimal solutions found by the algorithm and x a solution in the set. Smaller values of GD indicate that the set P is closer to the Pareto optimal front. That is, smaller values of GD mean better convergence of solutions.

$$GD = \underset{x \in P}{average} \left\{ \left[\sum_{i=1}^{m} (f_i(x))^2 \right]^{\frac{1}{2}} - 1 \right\} \tag{2}$$

To visually assess the distribution of solutions in objective space, we plot solutions projected to a two dimensional plane.

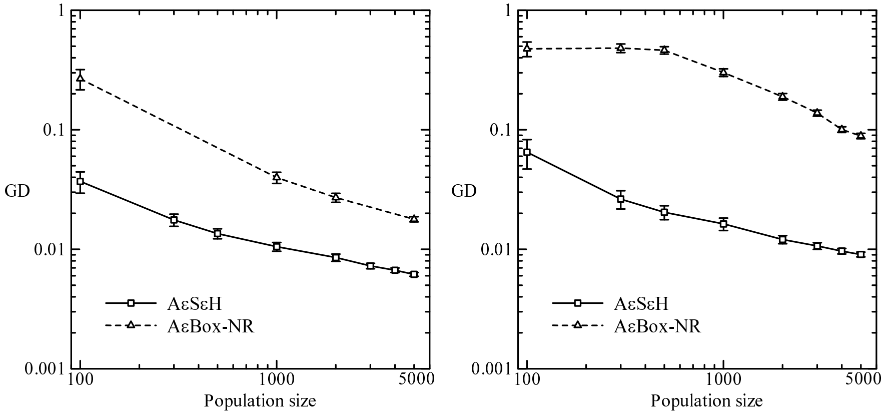
We run the algorithms 30 times and present average results, unless stated otherwise. We use a different random seed in each run, but all algorithms use the same seeds. The number of generations is set to 100 generations, and population size varies from to 100 to 20000, $|\mathcal{P}| = |\mathcal{Q}|$. As variation operators, the algorithms use SBX crossover and polynomial mutation, setting their distribution exponents to $\eta_c = 15$ and $\eta_m = 20$, respectively. Crossover rate is $pc = 1.0$, crossover rate per variable $pcv = 0.5$, and mutation rate per variable is $pm = 1/n$.

For A$\varepsilon_s\varepsilon_h$EMyO (A$\varepsilon_s\varepsilon_h$ for short) we set the reference neighborhood size H_{size}^{Ref} to 20 individuals. On the other hand, for Adaptive ε-Box with Neighborhood Recombination (AεBox-NR), we set the size of the neighborhood to 10% of the population size, a value that gave the best results in [6].

4 Simulation Results and Discussion

4.1 Convergence

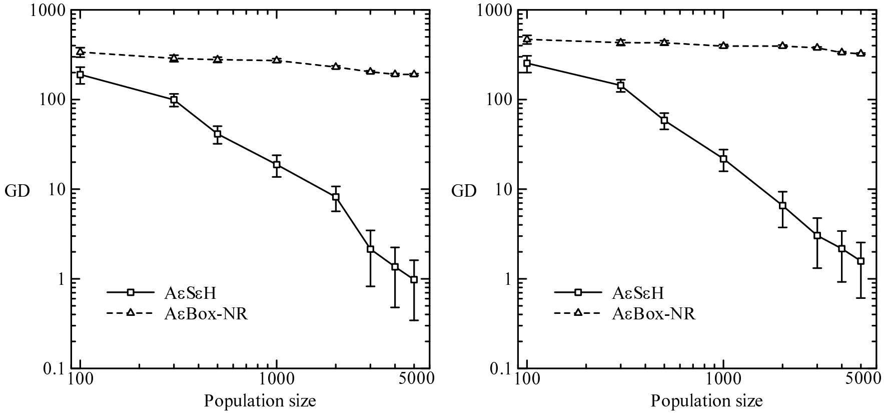
Fig.1 shows GD over population size by A$\varepsilon_s\varepsilon_h$ and AεBox-NR at generation $T = 100$ on problem DTLZ2 for $m = 5$ and $m = 6$ objectives. It can be seen



(a) $\boldsymbol{m = 5}$ objectives

(b) $\boldsymbol{m = 6}$ objectives

Fig. 1. GD after 100 generations for various population sizes, DTLZ2



(a) $\boldsymbol{m = 5}$ objectives

(b) $\boldsymbol{m = 6}$ objectives

Fig. 2. GD after 100 generations for various population sizes, DTLZ3

that smaller values of GD are achieved by using larger populations. Note that GD by the proposed A$\boldsymbol{\varepsilon_s\varepsilon_h}$ is substantially better than by AεBox-NR for any population size. A larger performance difference between the two algorithms can be seen by increasing the number of objectives from $\mathbf{5}$ to $\mathbf{6}$.

Similarly, **Fig.2** shows results for DTLZ3 problem. Analogous to problem DTLZ2, the proposed algorithm A$\boldsymbol{\varepsilon_s\varepsilon_h}$ performs better than AεBox-NR for any population size. In fact, in DTLZ3 the improvement in performance by the proposed algorithm is more notorious than in the case of DTLZ2. Note that increasing population size does not help much AεBox-NR, whereas GD by A$\varepsilon_s\varepsilon_h$ decreases in two orders of magnitude when population size is increased from $|\mathcal{P}| = 100$ to $|\mathcal{P}| = 5000$. However, it should be noticed that variance is larger and increases considerably with population size for the DTLZ3 problem. This is because convergence is harder in this problem and a larger number of generations are required to facilitate convergence of most individuals of the population towards the true Pareto front.

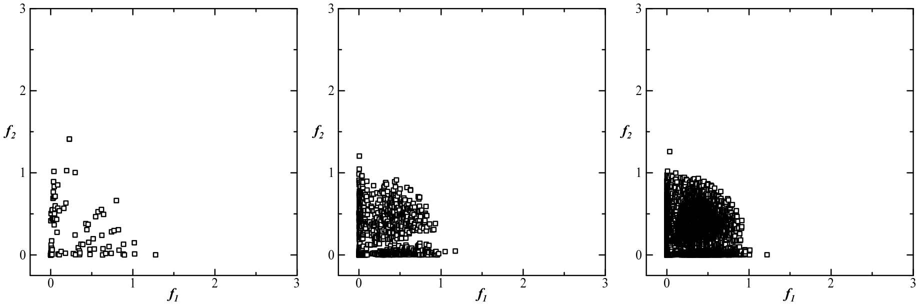
Results for DTLZ4 are similar to DTLZ2. Also, results on $m = 4$ objectives for all problems follow a similar trend to those observed on $m = 5$ and $m = 6$ objectives. Due to space limitations, those results are not included here.

4.2 Distribution

Fig.3 shows the f_1-f_2 objective values of the obtained non-dominated solutions by the proposed A$\varepsilon_s\varepsilon_h$ at generation 100. Results are shown for DTLZ2, $m = 6$ objectives, running the algorithm with some representative population sizes. Similarly, **Fig.4** shows solutions by AεBox-NR. In DTLZ2 the sum of the squares of the fitness values of a Pareto optimal solution is one. Thus, on a problem with more than 2 objectives the f_1-f_2 values of Pareto optimal solution fall within the positive quadrant of the circle of radius one. From **Fig.3** note that objective values are close to or within the positive quadrant of the circle of radius one. This is in accordance with the good convergence values observed for GD discussed above. Increasing the population size there is a better coverage of the quadrant, which implies a better distribution of solutions in objective space, and fewer solutions are located outside the quadrant. On the other hand, from **Fig.4** it can be seen that solutions by AεBox-NR tend to focus on extreme regions of objective space, where one or more objective values are close to 0, and many of them are far away from the optimal front. This effect reduces when a large population size is used, such as $|\mathcal{P}| = 5000$, but still there are many extreme solutions away from the optimal front and those located within the positive quadrant are not able to fully cover it.

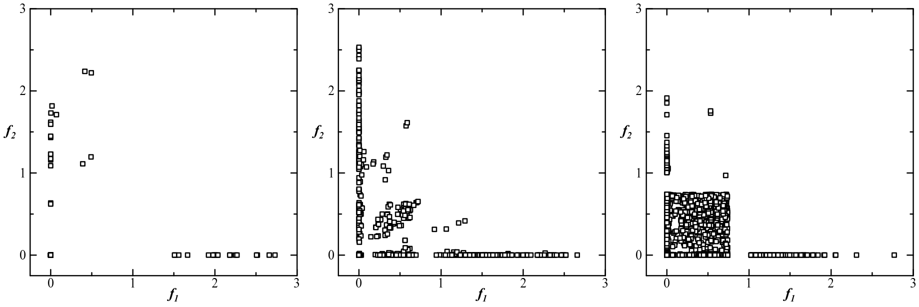
4.3 GD over the Generations and Larger Population Sizes

Fig.5 shows the transition of GD over the generations by A$\varepsilon_s\varepsilon_h$ on $m = 6$ objectives DTLZ2 and DTLZ3 problems, varying population sizes from $|\mathcal{P}| = 100$ to $|\mathcal{P}| = 20000$ individuals. Note that from early generations the algorithm with a larger population shows better convergence. This is a clear indication that population size is very important to support appropriately the evolutionary search on many-objective problems. On DTLZ2, note that for large populations initially there is a fast convergence, but after 50 generations or so the algorithm slows down significantly. On DTLZ3, convergence at the beginning is slower than on DTLZ2 but after some generations the effect of population size becomes more evident. Note that after 50 generations convergence speeds up significantly for populations $|\mathcal{P}| = 5000$ and $|\mathcal{P}| = 10000$ individuals. For



(a) $|\mathcal{P}| = \mathbf{100}$ individuals (b) $|\mathcal{P}| = \mathbf{1000}$ individuals (c) $|\mathcal{P}| = \mathbf{5000}$ individuals

Fig. 3. Obtained non-dominated solutions after 100 generations by A$\varepsilon_s\varepsilon_h$ for various population sizes, DTLZ2. Projection in plane $\boldsymbol{f_1}$-$\boldsymbol{f_2}$.



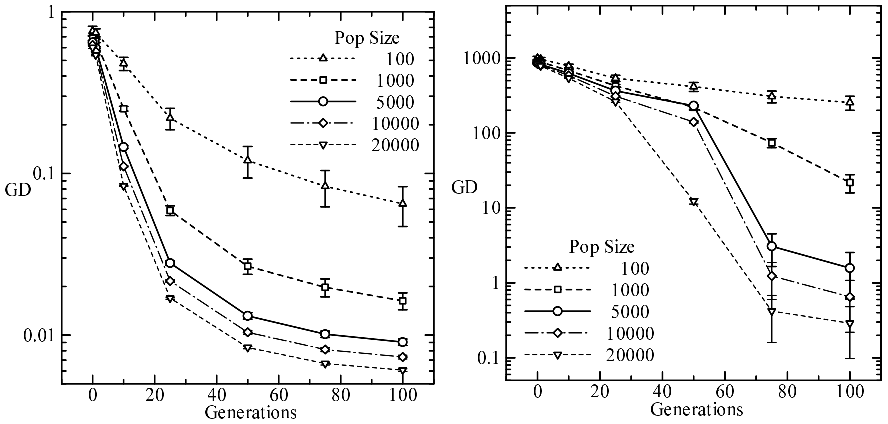
(a) $|\mathcal{P}| = \mathbf{100}$ individuals (b) $|\mathcal{P}| = \mathbf{1000}$ individuals (c) $|\mathcal{P}| = \mathbf{5000}$ individuals

Fig. 4. Obtained non-dominated solutions after 100 generations by AεBox-NR for various population sizes, DTLZ2. Projection in plane $\boldsymbol{f_1}$-$\boldsymbol{f_2}$

$|\mathcal{P}| = \mathbf{20000}$ individuals convergence speeds up earlier, at generation **20**. However, after generation **80** convergence slows down similar to DTLZ2. Although in both problems, DTLZ2 and DTLZ3, there is still room for converging closer to the optimal Pareto front the algorithm in final generations seems to stagnate. This suggests that the operators of variation themselves might need to be improved, particularly for the latest stage of the search when the population is approaching the Pareto optimal front. We would like to look into this in a future work.

4.4 Adaptation

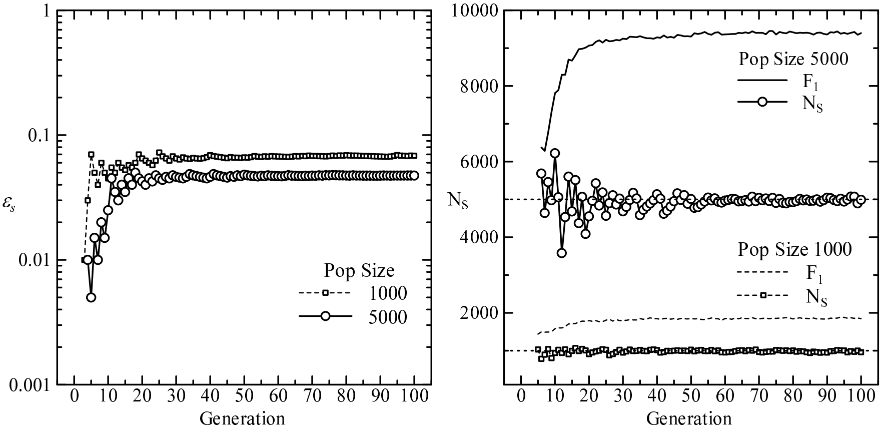
Fig.6 (a) shows the adaptation of ε_s for ε-sampling and **Fig.6 (b)** shows the number of solutions on the first front $\boldsymbol{\mathcal{F}_1}$ after non-dominated sorting together with the number of sampled solutions $\boldsymbol{N_S}$ by the ε-sampling procedure using



(a) DTLZ2 problem

(b) DTLZ3 problem

Fig. 5. GD over the generations for various population sizes, $\boldsymbol{m = 6}$ objectives

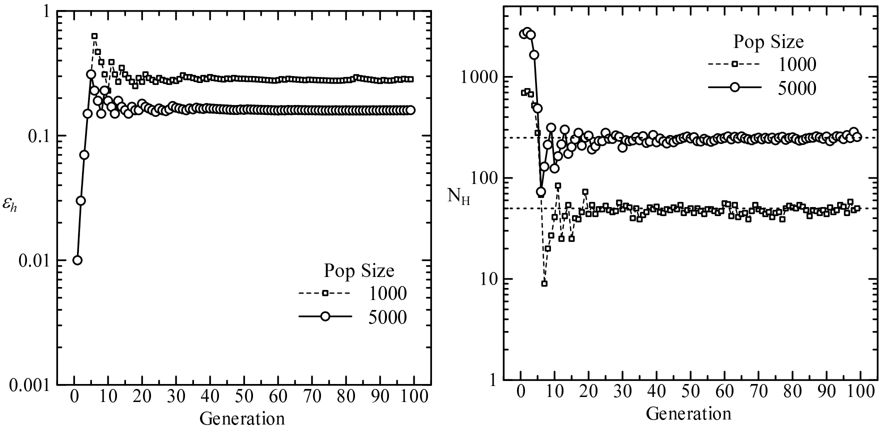


(a) $\boldsymbol{\varepsilon_s}$ adaptation

(b) $\boldsymbol{N_S}$ adaptation

Fig. 6. Adaptation for ε-sampling, DTLZ2, $\boldsymbol{m = 6}$ objectives

the adapted $\boldsymbol{\varepsilon_s}$ parameter. Results are shown for population sizes $|\boldsymbol{\mathcal{P}}| = \mathbf{1000}$ and $|\boldsymbol{\mathcal{P}}| = \mathbf{5000}$. From **Fig.6 (b)** note that $\boldsymbol{\mathcal{F}_1}$ is larger than $\boldsymbol{\mathcal{P}}$ since early generations and quickly approaches $2|\boldsymbol{\mathcal{P}}|$. The number of solutions $\boldsymbol{N_S}$ obtained after ε-sampling from $\boldsymbol{\mathcal{F}_1}$ is kept around the desired number $|\boldsymbol{\mathcal{P}}|$ thanks to the adaptation of $\boldsymbol{\varepsilon_s}$, as shown in **Fig.6 (a)**. From **Fig.6 (a)** note that $\boldsymbol{\varepsilon_s}$ is quickly adapted from its initial value so that $\boldsymbol{N_S}$ approaches the desired value $|\boldsymbol{\mathcal{P}}|$. Also note that the value that $\boldsymbol{\varepsilon_s}$ takes depends on the population size.



(a) ε_h adaptation (b) N_H adaptation

Fig. 7. Adaptation for neighborhood creation, DTLZ2, $m = 6$ objectives

Similar to the previous figure, **Fig.7 (a)** shows the adaptation of ε_h for neighborhood creation and **Fig.7 (b)** shows the number created neighborhoods N_H by the ε-hood creation procedure using the adapted ε_h parameter. From **Fig.7 (a)** note that ε_h is quickly adapted from its initial value, it takes values larger than ε_s as should be expected, and depends on population size. From **Fig.7 (b)** note that at the beginning the number of neighborhoods is quite large but thanks to the quick adaptation of ε_h the number of created neighborhoods N_H approaches the specified number N_H^{Ref}, $\mathbf{50}$ for $|\mathcal{P}| = \mathbf{1000}$ and $\mathbf{250}$ for $|\mathcal{P}| = \mathbf{5000}$.

The above results show that adaption is working properly for both ε-sampling and ε-hood creation.

5 Conclusions

This work has proposed an evolutionary many-objective optimizer that uses adaptive ε-sampling to select a subset of well distributed solutions for the surviving population. The method also uses adaptive ε-dominance to create neighborhoods of surviving solutions and performs mating between individuals of the same neighborhood to enhance the effectiveness of recombination. We verified the performance of the algorithm using DTLZ problems, observing the effects of increasing the population size on convergence and distribution of solution. We showed that for any population size the proposed method achieves substantially better quality of solutions in terms of convergence and distribution compared to Aε-Box with Neighborhood Recombination. We also showed that the method can successfully adapt the ε parameters used for truncation and neighborhood creation. In the future we would like to look into the operators of variation, aiming to further improve convergence. Also, we should analyze with more detail the impact of on performance of neighborhood size. In addition, we would like to test the proposed method on other classes of problems.

References

1. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
2. Coello, C., Van Veldhuizen, D., Lamont, G.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, Boston (2002)
3. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary Many-Objective Optimization: A Short Review. In: Proc. IEEE Congress on Evolutionary Computation (CEC 2008), pp. 2424–2431. IEEE Press (2008)
4. Sato, H., Aguirre, H., Tanaka, K.: Genetic Diversity and Effective Crossover in Evolutionary Many-objective Optimization. In: Coello, C.A.C. (ed.) LION 5. LNCS, vol. 6683, pp. 91–105. Springer, Heidelberg (2011)
5. Aguirre, H., Oyama, A., Tanaka, K.: Adaptive Epsilon-Ranking and Distribution Search on Evolutionary Many-objective Optimization. In: Proc. 2012 Italian Workshop on Artificial Life and Evolutionary Computation: Artificial Life, Evolution and Complexity (WIVACE 2012), CD-ROM, pp. 1–12 (2012)
6. Kowatari, N., Oyama, A., Aguirre, H., Tanaka, K.: A Study on Large Population MOEA Using Adaptive ϵ-Box Dominance and Neighborhood Recombination for Many-Objective Optimization. In: Hamadi, Y., Schoenauer, M. (eds.) LION 6. LNCS, vol. 7219, pp. 86–100. Springer, Heidelberg (2012)
7. Kowatari, N., Oyama, A., Aguirre, H., Tanaka, K.: Analysis on Population Size and Neighborhood Recombination on Many-Objective Optimization. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN XII, Part II. LNCS, vol. 7492, pp. 22–31. Springer, Heidelberg (2012)
8. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, KanGAL report 200001 (2000)
9. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. Evolutionary Computation 10(3), 263–282 (2002)
10. Aguirre, H., Tanaka, K.: Insights on Properties of Multi-objective MNK-Landscapes. In: Proc. 2004 IEEE Congress on Evolutionary Computation, pp. 196–203. IEEE Service Center (2004)
11. Aguirre, H., Tanaka, K.: Working Principles, Behavior, and Performance of MOEAs on MNK-Landscapes. European Journal of Operational Research 181(3), 1670–1690 (2007)
12. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Multi-Objective Optimization Test Problems. In: Proc. 2002 Congress on Evolutionary Computation, pp. 825–830. IEEE Service Center (2002)
13. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, Swiss Federal Institute of Technology, Zurich (1999)