Adaptive ϵ-Ranking and Distribution Search on 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@shinshu-u.ac.jp oyama@flab.isas.jaxa.jp ktanaka@shinshu-u.ac.jp

Abstract. In this work, we study the effectiveness of Adaptive ϵ-Ranking for distribution search in the context of many-objective optimization. Adaptive ϵ-Ranking re-classifies sets of non-dominated solutions using iteratively a randomized sampling procedure that applies ϵ-dominance with a mapping function $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}'(\boldsymbol{x})$ to bias selection towards the distribution of solutions implicit in the mapping. We analyze the effectiveness of Adaptive ϵ-Ranking with three linear mapping functions for ϵ-dominance and study the importance of recombination to properly guide the algorithm towards the distribution we seek to find. As test problems, we use functions of the DTLZ family with $M = 6$ objectives, varying the number of variables N from 10 to 50.

1 Introduction

Recently, there is a growing interest on applying multi-objective evolutionary algorithms (MOEAs) to solve many-objective optimization problems [1], where the number of objective functions to optimize simultaneously is more than three. Historically, most applications of MOEAs have dealt with two and three objective problems leading to the development of several evolutionary approaches that work successfully in these low dimensional objective spaces. However, it is well known that conventional MOEAs [2, 3] scale up poorly with the number of objectives of the problem. The poor performance of conventional MOEAs is attributed to an increased complexity inherent to high dimensional spaces, the use of operators of selection and variation that fail to take into account the characteristics of many-objective landscapes [4–6], and the use of population sizes inappropriate to support the evolutionary search on high dimensional spaces [7].

MOEAs seek to find trade-off solutions with good properties of convergence to the Pareto front, well spread, and well distributed along the front. A good distribution of solutions is usually assumed to be uniform. However, other distributions are often desired, either because of preferences or because they are required to extract relevant knowledge about the problem in order to provide useful guidelines to designers during the implementation of preferred solutions.

We focus on distribution search in the context of many-objective optimization, where in addition to good convergence towards the optimal Pareto front we

are required to find a set of trade-off solutions spread according to the distribution we seek to achieve. Methods based on relaxed forms of Pareto dominance, such as ϵ-dominance [8], can be used to search for a desired distribution of solutions. In this work, we adopt Adaptive ϵ-Ranking [9], a procedure that reclassifies sets of non-dominated solutions using iteratively a randomized sampling procedure that applies ϵ-dominance with a mapping function $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}^{'}(\boldsymbol{x})$ to bias selection towards the distribution implicit in the mapping. We analyze the effectiveness of Adaptive ϵ-Ranking with three linear mapping functions for ϵ-dominance and study the importance of recombination to properly guide the algorithm towards the distribution we seek to find. These mappings try to produce distributions of solutions evenly spaced, spaced by a distance that increases linearly from the center towards the extremes of objective space, and spaced by a distance that decreases linearly from the center towards the extremes. As test problems, we use functions of the DTLZ family [10] with $M = 6$ objectives, varying the number of variables N from 10 to 50. Our experiments reveal that in addition to setting a proper mapping function, in many-objective problems recombination plays a significant role to induce the distribution we aim to achieve.

2 Pareto dominance and ϵ-dominance

Let us consider, without loss of generality, a maximization multi-objective problem with M objectives:

$$maximize \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})) \tag{1}$$

where $\boldsymbol{x} \in \mathcal{S}$ is a solution vector in the feasible solution space $\mathcal{S}$, and $f_1(\cdot), f_2(\cdot), \cdots, f_M(\cdot)$ the M objectives to be maximized.

Pareto dominance is widely used to rank solutions in multi-objective optimization. It can be defined as follows.

Pareto dominance. A solution $\boldsymbol{x}$ is said to Pareto dominate other solution $\boldsymbol{y}$, denoted by $\boldsymbol{f}(\boldsymbol{x}) \succeq \boldsymbol{f}(\boldsymbol{y})$, if the two following conditions are satisfied:

$$\begin{array}{ll} \forall i \in \{1, \cdots, M\} & f_i(\boldsymbol{x}) \geq f_i(\boldsymbol{y}) \ \wedge \\ \exists i \in \{1, \cdots, M\} & f_i(\boldsymbol{x}) > f_i(\boldsymbol{y}). \end{array} \tag{2}$$

In addition, relaxed forms of Pareto dominance are used to maintain an archive of non-dominated solutions and also within the selection operator to rank solutions during evolution. One way to implement a relaxed form of dominance is ϵ-dominance [8]. In ϵ-dominance, the objective vector $\boldsymbol{f}(\boldsymbol{x})$ of a solution $\boldsymbol{x}$ is first mapped to another point $\boldsymbol{f}^{'}(\boldsymbol{x})$ in objective space and dominance is calculated using the mapped point. It can be defined as follows

ϵ-dominance. A solution $\boldsymbol{x}$ is said to ϵ-dominate other solution $\boldsymbol{y}$, denoted by $\boldsymbol{f}(\boldsymbol{x}) \succeq^{\epsilon} \boldsymbol{f}(\boldsymbol{y})$, if the following conditions are satisfied:

$$\begin{array}{ll} \boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}^{'}(\boldsymbol{x}) & \\ \forall i \in \{1, \cdots, M\} & f_i^{'}(\boldsymbol{x}) \geq f_i(\boldsymbol{y}) \ \wedge \\ \exists i \in \{1, \cdots, M\} & f_i^{'}(\boldsymbol{x}) > f_i(\boldsymbol{y}), \end{array} \tag{3}$$

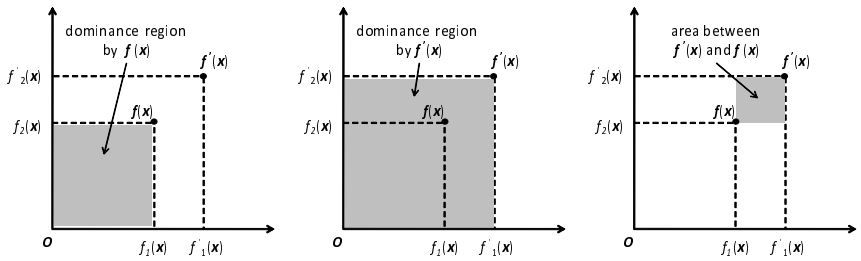
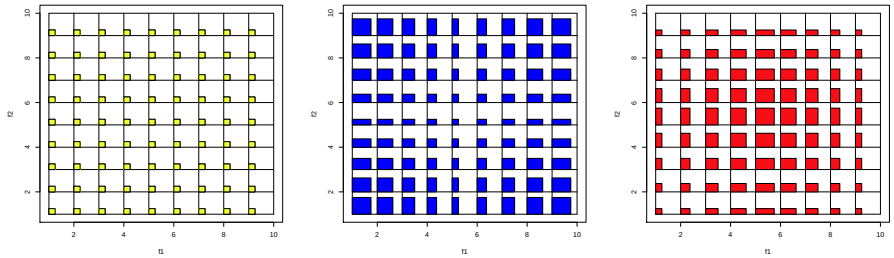


Fig. 1. A point $\boldsymbol{f}(\boldsymbol{x})$ in objective space, its corresponding mapped point $\boldsymbol{f}'(\boldsymbol{x})$, their dominance regions and the area between $\boldsymbol{f}'(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x})$.



(a) Additive (b) Linear Expansion (c) Linear Contraction

Fig. 2. Areas between $\boldsymbol{f}(\boldsymbol{x})$ and its mapped point $\boldsymbol{f}'(\boldsymbol{x})$ for several points by Additive, Expansion, and Contraction using the same ϵ in the three mapping functions.

where $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}'(\boldsymbol{x})$ is a mapping function that depends on a parameter ϵ.

Fig.1 illustrates a point $\boldsymbol{f}(\boldsymbol{x})$ and its mapped point $\boldsymbol{f}'(\boldsymbol{x})$ in objective space by a mapping function $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}'(\boldsymbol{x})$, their dominance regions and the area between $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}'(\boldsymbol{x})$. Note that a larger area between $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}'(\boldsymbol{x})$ would correspond to a larger expansion of $\boldsymbol{f}'(\boldsymbol{x})$'s dominance region. Various kinds of mapping functions have been used for ϵ-dominance, such as multiplicative, additive, and logarithmic. In the next section we describe three additive mapping functions used in our study.

3 Mapping functions

In this work we investigate distribution search using three additive mapping functions for ϵ-dominance in the context of many-objective optimization. The mappings are designed to induce distributions of solutions evenly spaced by ϵ, spaced by a distance that increases linearly from the center towards the extremes of objective space, and spaced by a distance that decreases linearly from the center towards the extremes. These mappings are called Additive, Linear Expansion from the Center, and Linear Contraction from the Center, respectively. **Fig.2** illustrates the areas between $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}'(\boldsymbol{x})$ for several points in objective space by these mappings. Their definitions are as follows.

Additive. Additive maps $\boldsymbol{f}(\boldsymbol{x})$ to $\boldsymbol{f}^{'}(\boldsymbol{x})$ by adding the same value ϵ to all coordinates f_i, independently of the position of $\boldsymbol{f}(\boldsymbol{x})$ in objective space. The expression for Additive mapping is as follows

$$f_i^{'}(\boldsymbol{x}) = f_i(\boldsymbol{x}) + \epsilon, \quad i = 1, \cdots, m \tag{4}$$

Linear Expansion from Center. Linear Expansion maps $\boldsymbol{f}(\boldsymbol{x})$ to $\boldsymbol{f}^{'}(\boldsymbol{x})$ by adding the same value ϵ to all coordinates f_i and an additional δ_i value that increases linearly with the distance of f_i to a central point. The expression for Linear Expansion from Center is as follows

$$f_i^{'}(\boldsymbol{x}) = f_i(\boldsymbol{x}) + \epsilon + \delta_i, \quad i = 1, \cdots, m \tag{5}$$

where δ_i is calculated by

$$\delta_i = \gamma \frac{|f_i(\boldsymbol{x}) - \bar{f}_i|}{\frac{1}{2}|f_i^{max} - f_i^{min}|}. \tag{6}$$

Here, $\gamma > 0.0$ is a parameter that controls the slope of the linear increase, f_i^{max} and f_i^{min} are the maximum and minimum values in objective f_i, calculated from all solutions in the population, and $\bar{f}_i$ is the central point calculated by

$$\bar{f}_i = \frac{1}{2}(f_i^{max} + f_i^{min}). \tag{7}$$

Linear Contraction from Center. Linear Contraction, similar to Linear Expansion, also adds the same ϵ to all coordinates f_i and a variable δ_i to map $\boldsymbol{f}(\boldsymbol{x})$ to $\boldsymbol{f}^{'}(\boldsymbol{x})$ using eq.(5). However, here δ_i is calculated so that its value decreases linearly from the central point. In case $f_i(\boldsymbol{x}) > \bar{f}_i$, δ_i is calculated by

$$\delta_i = \gamma \frac{|f_i(\boldsymbol{x}) - f_i^{max}|}{\frac{1}{2}|f_i^{max} - f_i^{min}|} \tag{8}$$

Otherwise, δ_i is calculated by

$$\delta_i = \gamma \frac{|f_i(\boldsymbol{x}) - f_i^{min}|}{\frac{1}{2}|f_i^{max} - f_i^{min}|} \tag{9}$$

4 Adaptive ϵ-Ranking (AϵR)

Adaptive ϵ-Ranking (AϵR) [9] re-classifies sets of non-dominated solutions using iteratively a randomized sampling procedure that applies ϵ-dominance [8] with mapping function $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}^{'}(\boldsymbol{x})$ to favor the distribution of solutions implicit in the mapping. In previous works, we have used AϵR with a multiplicative mapping function for ϵ-dominance. In this work we implement AϵR within the NSGA-II framework [11] and use three different additive mapping functions, as mentioned above.

AϵR in the framework of NSGA-II assigns a new primary ranking of solutions by reclassifying the fronts $\mathcal{F}_i$ $(i = 1, \cdots, N_F)$ found by non-domination sorting into fronts $\mathcal{F}^{\epsilon} = \{\mathcal{F}_j^{\epsilon}\}$ $(j = 1, 2, \cdots, N_F^{\epsilon})$, where $N_F^{\epsilon} \geq N_F$. The main steps are as follows:

Step 1 Assign the first front for reclassification, $\mathcal{A} = \mathcal{F}_1$. Set front counters $i = 1$ and $j = 1$.
Step 2 Call ϵ-sampling with mapping function $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}^{'}(\boldsymbol{x})$ and parameter $\epsilon > 0.0$ to get a sample $\mathcal{S}$ from $\mathcal{A}$ and their ϵ-dominated solutions $\mathcal{D}^{\epsilon}$.
Step 3 Assign the sampled solutions as the reclassified front, $\mathcal{F}_j^{\epsilon} = \mathcal{S}$.
Step 4 If there are still fronts to reclassify, $i + 1 < N_F$, join the ϵ-dominated solutions with solutions of the next front and assign them for reclassification, $\mathcal{A} = \mathcal{D}^{\epsilon} \cup \mathcal{F}_{i+1}$. Otherwise, $\mathcal{A} = \mathcal{D}^{\epsilon}$.
Step 5 Update counters $i = i + 1$, $j = j + 1$. If $\mathcal{A}$ is not empty, go to **Step 2**.

The sampling procedure referred in **Step 2** applies ϵ-dominance with mapping function $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}^{'}(\boldsymbol{x})$ and parameter $\epsilon > 0.0$ to virtually extend the dominance area of the sampled solution in order not to include closely located solutions in the sample. Thus, solutions in the sample are expected to be spread according to the mapping function. In other words, AϵR tries to search the distribution of solutions specified implicitly by the mapping function. The ϵ-sampling procedure is as follows:

Step 1 Select extreme solutions from $\mathcal{A}$ (without replacement) and assign them to the sample set $\mathcal{S}$.
Step 2 Select randomly one solution from $\mathcal{A}$ (without replacement) and assign it to $\mathcal{S}$.
Step 3 Remove from $\mathcal{A}$ solutions ϵ-dominated by the randomly selected solution and assign them to $\mathcal{D}^{\epsilon}$.
Step 4 If $\mathcal{A}$ is not empty go to **Step 2**.

The number of rank-1 solutions $|\mathcal{F}_1^{\epsilon}|$ after reclassification depends on the value set to ϵ (≥ 0). Larger values of ϵ imply that sampled solutions ϵ-dominate larger areas, increasing the likelihood of having more ϵ-dominated solutions excluded from the sample that form $\mathcal{F}_1^{\epsilon}$. AϵR adapts ϵ at each generation so that $|\mathcal{F}_1^{\epsilon}|$ is close to the size of the parent population $|\mathcal{P}|$ with the following rule. If $|\mathcal{F}_1^{\epsilon}| > |\mathcal{P}|$ it increases the step of adaptation $\Delta \leftarrow \min(\Delta \times 2, \Delta_{max})$ and $\epsilon \leftarrow \epsilon + \Delta$. Otherwise, if $|\mathcal{F}_1^{\epsilon}| < |\mathcal{P}|$ 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 $\Delta_{min} = 0.0001$.

5 Simulation Results and Discussion

5.1 Test Problems and Experimental Setup

We study the performance and behavior of NSGA-II and AϵR on the DTLZ test functions family [10]. These functions are scalable in the number of objectives and variables and thus allow for a many-objective study. In this work we present results for DTLZ2 with $M = 6$ objectives varying the total number of variables N from 10 to 50. DTLZ2 has a non-convex Pareto-optimal surface in the first quadrant of the M-dimensional unit hyper-sphere. For Linear Expansion and Contraction in AϵR we vary the parameter γ from 0.10 to 0.30, which

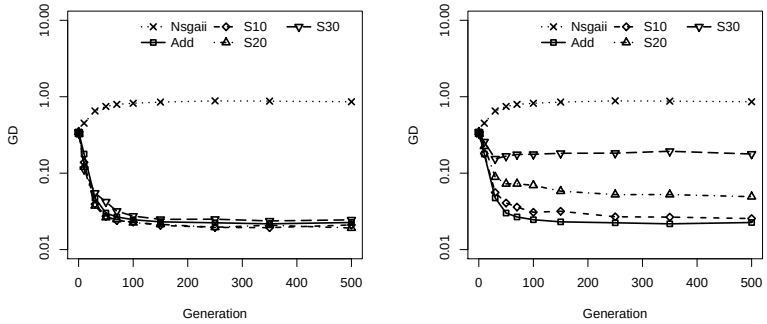
determines the slope of the linear expansion or contraction, respectively. We run the algorithms 30 times and present average results, unless stated otherwise. We use a different random seed in each run, but all algorithms use the same seeds. The number of generations is set to 500 generations, and population size to 300 ($\mathcal{P} = \mathcal{Q} = 300$). 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 $pcv = \{0.5, 0.1\}$, and mutation rate per variable is $pm = 1/N$.

5.2 Effects on Convergence

In this section we analyze convergence by conventional NSGA-II and AϵR with different mappings. **Fig.3** shows the average Generational Distance (GD) over the generations setting the algorithms with $pcv = 0.5$, a widely used value for crossover rate per variable. Results are shown for the DTLZ2 function with $N = \{10, 30, 50\}$ variables. In each plot results by NSGA-II and AϵR with Additive mapping (labeled in the plots as Add) are included for reference, together with results by AϵR with Expansion or AϵR with Contraction using slope values of $\gamma = 0.10$, $\gamma = 0.20$ and $\gamma = 0.30$ (labeled in the plots as S10, S20 and S30, respectively). From this figure, note that GD by NSGA-II increases since the first generation, for any number of variables N, and remains higher than the GD of the random initial population. In other words, NSGA-II diverges rather than converges towards the true Pareto front. On the contrary, GD by AϵR algorithms reduces with the number of generations and becomes significantly smaller compared to GD of the initial population. That is, sampling using ϵ-dominance helps the algorithm to converge towards the true Pareto front. These results are in accordance with previous reports on ϵ-dominance based algorithms by several researches.

The bad performance of NSGA-II can be explained in terms of selection as follows. In many-objective problems most solutions are non-dominated since early generations. Thus, NSGA-II's selection relies mostly on the diversity estimation operator, which induces the population to spread rather than to converge. This misleads the algorithm towards dominance resistant solutions in the extreme regions of the Pareto front, where fitness of solutions take values close to 0.0 in some objectives and very large values in other objectives.

Looking at the three variants of AϵR, it can be seen that overall GD by Additive mapping is better than by the two other variants. However, note that the best value of the achieved GD gets worse as we increase the number of variables. This is mainly because it is more challenging to optimize a larger number of variables and also because when we increase the number of variables evolution starts from an initial population that is farther away from the true Pareto front, as evidenced by the larger values of GD at generation $t = 0$. Comparing Expansion and Additive mappings, we can see that GD's transition over the generations looks similar for $N = 10$ variables, but GD by Expansion becomes worse than by Additive for $N = 30$ and $N = 50$ variables. Note that GD by Expansion gets worse conform we increase the slope γ. In addition, note



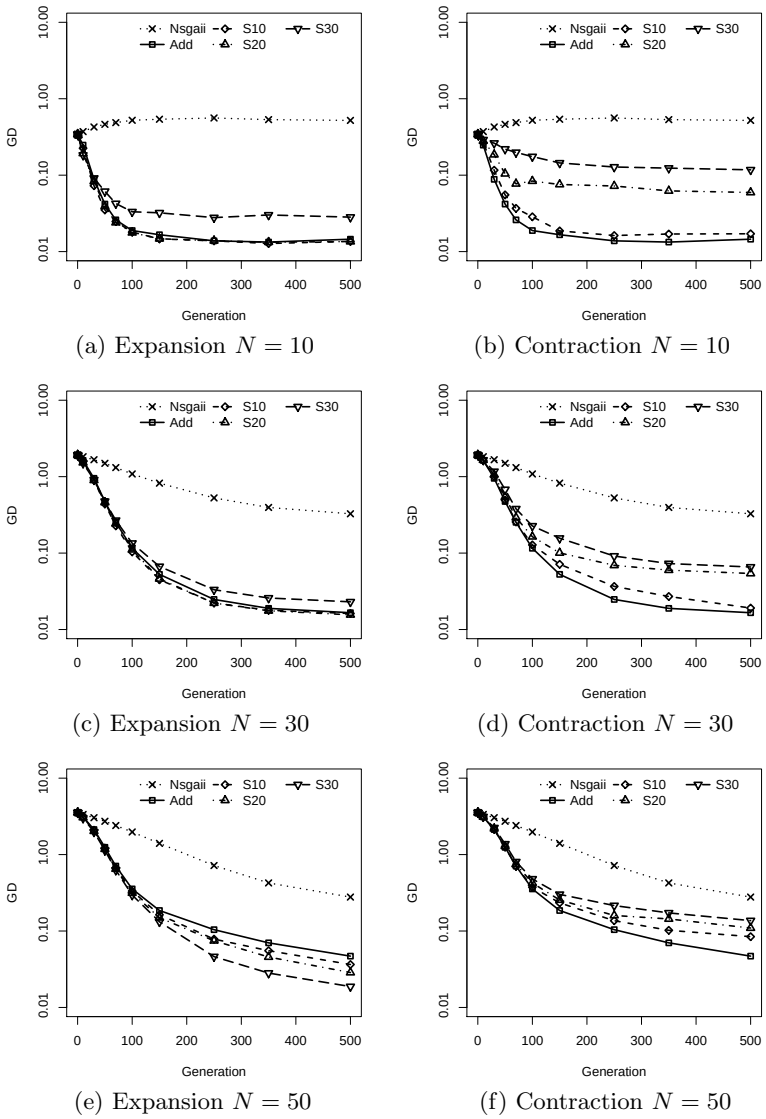
(a) Expansion $N = 10$ (b) Contraction $N = 10$

(c) Expansion $N = 30$ (d) Contraction $N = 30$

(e) Expansion $N = 50$ (f) Contraction $N = 50$

Fig. 3. Generational Distance by NSGA-II, Additive Epsilon, Linear Expansion and Linear Contraction. DTLZ2, $M = 6$, $N = \{10, 30, 50\}$, $pcv = 0.5$.

that for $N = 50$, after an initial stage in which GD improves (smaller values) we can see that GD starts to get worse (larger values). Comparing Contraction and Additive mappings, we can see that GD by Contraction with larger slope γ is worse than GD by Additive for small N. However, for $N = 50$ we can see that GD by Contraction becomes better than by Additive and also that larger values of the slope γ lead to better values of GD.



(a) Expansion $N = 10$ (b) Contraction $N = 10$

(c) Expansion $N = 30$ (d) Contraction $N = 30$

(e) Expansion $N = 50$ (f) Contraction $N = 50$

Fig. 4. Generational Distance by NSGA-II, Additive Epsilon, Linear Expansion and Linear Contraction. $M = 6$, $N = \{10, 30, 50\}$, $pcv = 0.1$.

5.3 Convergence Reducing Recombination Rate per Variable

Recent works on combinatorial problems have shown that diversity in variable space of non-dominated solutions gets significantly larger with the number of objectives and recombination may become too disruptive, affecting its effectiveness to find better solutions [6]. In this section we analyze the effects on convergence of a less explorative recombination by reducing the crossover rate

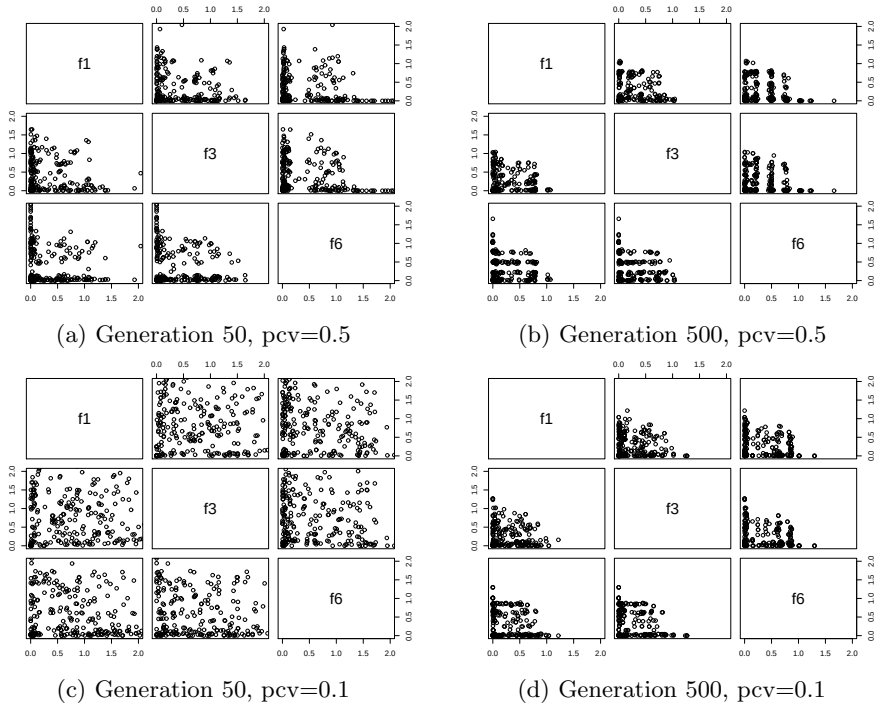
per variable from $pcv = 0.5$ to $pcv = 0.1$. Our intention is to verify whether our findings on combinatorial problems hold on continuous problems and analyze the relationship between the effectiveness of the recombination operator and the distribution of solutions that selection tries to induce. **Fig.4** shows the average GD by all algorithms with $pcv = 0.1$. From this figure, note the convergence trend by NSGA-II for $N = 30$ and $N = 50$ variables, which is not observed for $pcv = 0.5$. This is an indication that recombination with $pcv = 0.1$ is more effective than $pcv = 0.5$. This also implies that ineffective recombination is also responsible for the divergence of the algorithm towards dominance resistant solutions, and not only selection as suggested in the previous section. Looking at the AϵR algorithms, we can see that they can achieve better values of GD with $pcv = 0.1$ than with $pcv = 0.5$. Note also that the improvement of GD becomes more significant on problems with larger number of variables. The trends of GD by Expansion and Contraction mappings using $pcv = 0.1$ are similar to those observed for $pcv = 0.5$ for problems with up to 30 variables. However, in the case of $N = 50$ variables, note that Expansion achieves better GD than Additive and Contraction achieves worse GD than Additive. It is interesting to note that exactly the opposite trend is observed for $pcv = 0.5$ as shown in **Fig.3** (e) and (f). The overall better GD obtained by reducing recombination rate per variable from $pcv = 0.5$ to $pcv = 0.1$ suggests that the effectiveness of recombination is an issue that needs to be carefully considered in continuous problems.

5.4 Obtained solutions and their distribution

In previous sections we analyzed the effects of AϵR algorithms and rate of recombination per variable on the convergence of the algorithms. In this section we focus our analysis on the distributions of the obtained solutions.

Fig.5 shows scatter plots of the Pareto optimal solutions found by Additive mapping for one run of the algorithm. Results are shown in the range $[0.0, 2.0]$ for the planes formed by f_1, f_3 and f_6 at generations 50 and 500 using $pcv = 0.5$ and $pcv = 0.1$. On the other hand, **Fig.6** shows hexagon binning plots of solutions in the f_1-f_3 plane at generation 500. A hexagon binning is a form of bivariate histogram. A grid of hexagons is formed in the plane f_1-f_3 and the number of solutions falling in each hexagon are counted. The hexagons with count > 0 are plotted varying the radius of the hexagon in proportion to the counts. Results are shown in the range $[0.0, 1.25]$ for the three variants Additive, Contraction, and Expansion mappings using $pcv = 0.5$ and $pcv = 0.1$.

From **Fig.5**, note the different distributions of solutions obtained by using $pcv = 0.5$ and $pcv = 0.1$. At generation 50, we can observe that many solutions have become dominant resistant, i.e. f_1 and (or) f_3 approach 0.0, when $pcv = 0.5$. On the other hand, many more solutions are observed in the central regions of objective space when $pcv = 0.1$. At generation 500, most solutions have converged close to the optimum hypersphere of radius 1. However, solutions obtained by $pcv = 0.1$ are more evenly distributed than solutions obtained by $pcv = 0.5$. This is better illustrated by hexagon binning in **Fig.6** (a) and (b). For example, note that in the case of $pcv = 0.5$ there are 47 solutions, out of a



(a) Generation 50, pcv=0.5

(b) Generation 500, pcv=0.5

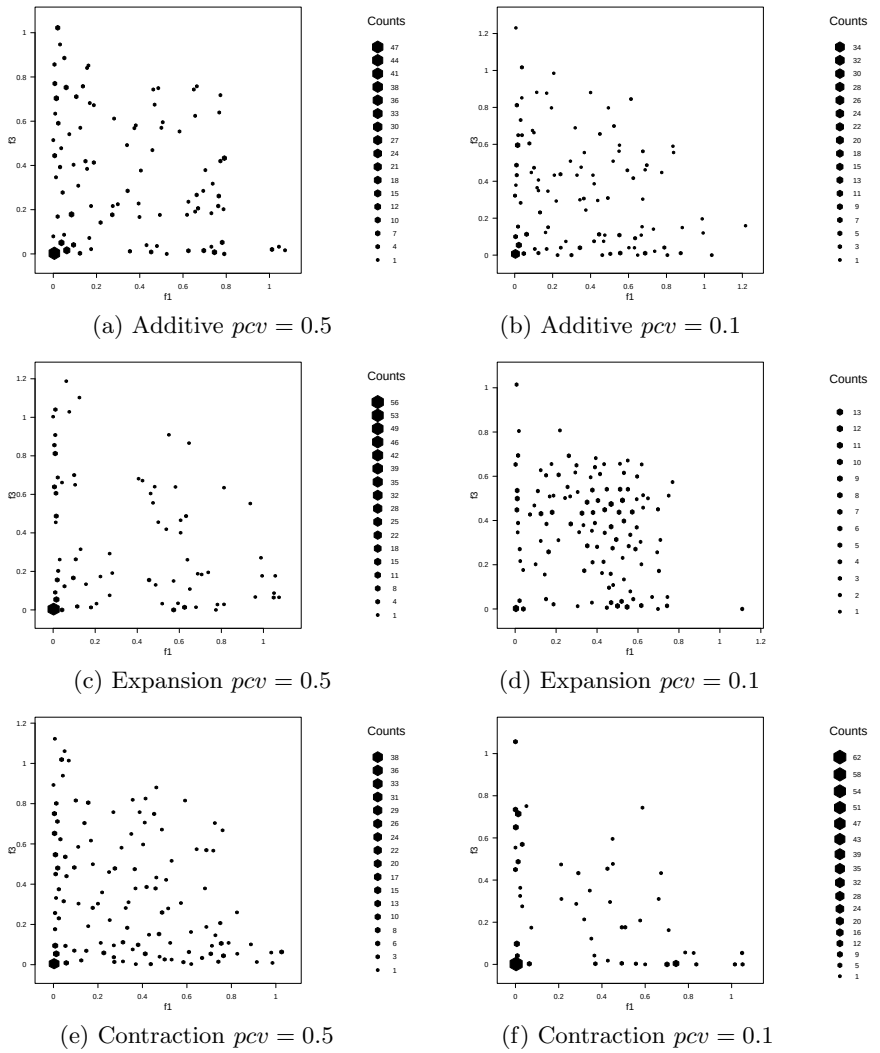
(c) Generation 50, pcv=0.1

(d) Generation 500, pcv=0.1

Fig. 5. Nondominated solutions by Additive mapping, range $[0.0, 2.0]$. DTLZ2, $M = 6$, $N = 50$.

maximum of 300, located close to $f_1 = f_3 = 0.0$, whereas there are 34 in the case of $pcv = 0.1$. The distribution of solutions by Additive mapping with $pcv = 0.1$ reflects more precisely the kind of distribution we want to achieve with the additive mapping described in section 3. In our analysis of GD it became apparent that an appropriate recombination rate is important to improve convergence. The analysis of the distribution of solutions suggests that selection alone, without considering a proper recombination, cannot induce the distribution we aim for.

Looking at the hexagon binning plots by Expansion and Contraction mappings, the relevance of recombination to achieve the desired distribution of solutions becomes clearer. Note from **Fig.6** (d) that in the case of $pcv = 0.1$ solutions tend to cluster around $f_1 = 0.5$, $f_3 = 0.5$, and $f_1 = f_3 = 0.5$, which is what should be expected for the Expansion mapping. However, that is not the case when $pcv = 0.5$, where a large number of solutions are clustered at $f_1 = f_3 = 0.0$, as shown in **Fig.6** (c). Similarly, note from **Fig.6** (f) that in the case of $pcv = 0.1$ there is a large number of solutions concentrated around $f_1 = f_3 = 0.0$ and along the axis as expected for the Contraction mapping. On the other hand, in the case of $pcv = 0.5$ the distribution of solutions in the objective space looks more uniform, which is counterintuitive for the Contraction mapping, as shown in **Fig.6** (e). In fact, it is interesting to note that the distribution by Contraction



(a) Additive $pcv = 0.5$ (b) Additive $pcv = 0.1$

(c) Expansion $pcv = 0.5$ (d) Expansion $pcv = 0.1$

(e) Contraction $pcv = 0.5$ (f) Contraction $pcv = 0.1$

Fig. 6. Hexagon binning of nondominated solutions in plane f_1-f_3 by Additive, Expansion and Contraction. DTLZ2, $M = 6$, $N = 50$, $pcv = 0.5$ and $pcv = 0.1$.

mapping with $pcv = 0.5$ shown in **Fig.6** (e) looks more uniform that the one by Additive mapping with $pcv = 0.5$ shown in **Fig.6** (a). Looking at **Fig.3** (f) we can also see that Contraction mapping with $pcv = 0.5$ achieves better GD than Additive mapping with $pcv = 0.5$. If GD would be our only criterion to evaluate the algorithms, it would seem that Contraction mapping works better than Additive mapping. However, the distribution rendered by Contraction is not what should be expected from such mapping. In the future we should analyze with more detail the reasons why Contraction mapping renders a more uniform distribution than Additive mapping when a large crossover rate is used.

6 Conclusions

In this work we have studied the effectiveness of Adaptive ϵ-Ranking for distribution search in the context of many-objective optimization. We have analyzed three additive mapping functions for the ϵ-sampling procedure of Adaptive ϵ-Ranking in order to bias the search towards different distributions of solutions. We also analyzed the relationship between the effectiveness of the recombination operator and the distribution of solutions that selection tries to induce. We have verified that in many-objective continuous problems a less explorative recombination can increase substantially convergence of solutions. Also, we verified that selection alone without considering a proper recombination rate cannot induce the distribution we seek to achieve. In the future, we would like to extend our analysis to other problems and look deeper into the relationship between recombination and selection for distribution search.

References

1. H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary Many-Objective Optimization: A Short Review", *In Proc. 2008 IEEE Congress on Evolutionary Computation*, IEEE Press, pp.2424-2431, 2008.
2. K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester, West Sussex, England, 2001.
3. C. Coello, D. Van Veldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, Boston, 2002.
4. 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.
5. H. Aguirre and K. Tanaka, "Working Principles, Behavior, and Performance of MOEAs on MNK-Landscapes", *European Journal of Operational Research*, Elsevier, vol. 181(3), pp. 1670-1690, Sep. 2007.
6. H. Sato, H. Aguirre and K. Tanaka, "Genetic Diversity and Effective Crossover in Evolutionary Many-objective Optimization", *Proc. Learning and Intelligent Optimization Conference (LION 5)*, Lecture Notes in Computer Science (Springer),
7. N. Kowatari, A. Oyama, H. Aguirre and K. Tanaka, "A Study on Large Population MOEA Using Adaptive Epsilon-Box Dominance and Neighborhood Recombination for Many-objective Optimization", *Proc. Learning and Intelligent Optimization Conference (LION 6)*, Lecture Notes in Computer Science (Springer), Jan. 2012.
8. M. Laumanns, L. Thiele, K. Deb and E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multi-objective Optimization", *Evolutionary Computation*, Vol.10, No.3, pp.263-282, Fall 2002.
9. H. Aguirre, K. Tanaka, "Adaptive ε-Ranking on Many-Objective Problems", *Evolutionary Intelligence*, Springer, Vol.2, No.4, pp.183-206, Dec. 2009.
10. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimization Test Problems". *Proc. 2002 Congress on Evolutionary Computation*, IEEE Service Center, pp.825–830, 2002.
11. 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.