



HAL
open science

Recombination and Self-Adaptation in Multi-objective Genetic Algorithms

Bruno Sareni, Jérémie Regnier, Xavier Roboam

► **To cite this version:**

Bruno Sareni, Jérémie Regnier, Xavier Roboam. Recombination and Self-Adaptation in Multi-objective Genetic Algorithms. Lecture Notes in Computer Science, 2004, vol. 2936, pp.115-126. 10.1007/978-3-540-24621-3_10 . hal-00766887

HAL Id: hal-00766887

<https://hal.science/hal-00766887>

Submitted on 19 Dec 2012

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.

Recombination and Self-Adaptation in Multi-objective Genetic Algorithms

Bruno Sareni, Jérémie Regnier, and Xavier Roboam

Laboratoire d'Electrotechnique et d'Electronique Industrielle
Unité mixte de Recherche INPT-ENSEEIHHT / CNRS
BP 7122 – 2 rue Camichel – 31071 Toulouse Cedex 7 – France
{sareni, regnier, roboam}@leei.enseeiht.fr,
<http://www.leei.enseeiht.fr>

Abstract. This paper investigates the influence of recombination and self-adaptation in real-encoded Multi-Objective Genetic Algorithms (MOGAs). NSGA-II and SPEA2 are used as example to characterize the efficiency of MOGAs in relation to various recombination operators. The blend crossover, the simulated binary crossover and the breeder genetic crossover are compared for both MOGAs on multi-objective problems of the literature. Finally, a self-adaptive recombination scheme is proposed to improve the robustness of MOGAs.

1 Introduction

In recent years, extensive research has been done in the field of Multi-Objective Genetic Algorithms (MOGAs) [1], [2], [3], [4]. However, most of works have been focused on selection, elitism and niching operators. Although the need of studying the influence of recombination and self-adaptation in MOGAs has been underlined [5], only a few contributions on these issues have been carried out [2], [6]. In this paper, we investigate the efficiency of three crossover operators for real-encoded MOGAs and propose a self-adaptive recombination scheme which improves their robustness. The second versions of the Non-dominated Sorting Genetic Algorithm (NSGA-II) [3] and of the Strength Pareto Genetic Algorithm (SPEA2) [4] are used as example to characterize the efficiency of the studied recombination operators on test problems of the literature.

2 Elitist Multi-objective Genetic Algorithms

Since the mid-1990s, there has been a growing interest in solving multi-objective problems by Genetic Algorithms. In particular, elitist MOGAs based on Pareto approaches have become more and more popular because of their capabilities to approximate the set of optimal trade-offs in a single run [1], [3], [4]. Elitist MOGAs use

an external population, namely *archive*, which preserves non-dominated individuals in the population. At each generation, individuals (*parents*) selected from the archive (and/or from the population) following Pareto domination rules are crossed and mutated to create new individuals (*children*). The population of children and the archive are merged to assess the non-dominated set of the next generation. If the number of non-dominated individuals is higher than the size of the archive, a *clustering* method is used to preserve most representative solutions and eliminate others in order to keep a constant archive size. Note that niching is used in the selection scheme when individuals involved in a tournament have the same Pareto domination rank.

The second version of the Non-dominated Sorting Genetic Algorithm (NSGA-II) is based on the principles previously exposed. NSGA-II determines all successive fronts in the population (the best front corresponding to the non-dominated set). Moreover, a *crowding distance* is used to estimate the density of solutions surrounding each individual on a given front. In a tournament, if individuals belong to the same front, the selected one is that with the greater crowding distance. This niching index is also used in the clustering operator to uniformly distribute the individuals on the Pareto front. All details of the algorithm can be found in [3].

The new version of the Strength Pareto Genetic Algorithm (SPEA2) is rather similar to the NSGA-II. It essentially differs in the clustering method used to merge non-dominated individuals in the archive and on the selection method based on a fitness assignment (called strength) related to Pareto ranking of individuals in the population [4].

In this, study NSGA-II and SPEA2 are taken as reference to investigate the influence of recombination and self-adaptation in real-encoded MOGAs.

3 Recombination and Self-Adaptive Procedures

We examine various recombination and self-adaptive procedures for real-encoded MOGAs :

3.1 The Blend Crossover

From two parent solutions $p_1(i)$ and $p_2(i)$, the blend crossover (BLX- α) creates one child $c(i)$ as follows [7] :

$$c(i) = p_1(i) + \beta(p_2(i) - p_1(i)) \quad (1)$$

where β is a random variable in the interval $[-\alpha, 1 + \alpha]$, i denoting the index related to the object variable of the child and parents solutions. If α is set to zero, this crossover creates a random solution inside the range defined by the parents similarly to the arithmetical crossover [8]. Eshelman and Schaffer have reported that BLX-0.5 (with $\alpha = 0.5$) performs better than BLX with any other α value in a number of test problems.

3.2 The Simulated Binary Crossover

The simulated binary crossover (SBX) simulates the working principle of the single point crossover operator on binary strings [9]. From two parent solutions $p_1(i)$ and $p_2(i)$, it creates two children $c_1(i)$ and $c_2(i)$ as follows :

$$\begin{cases} c_1(i) = 0.5[(1 + \beta)p_1(i) + (1 - \beta)p_2(i)] \\ c_2(i) = 0.5[(1 - \beta)p_1(i) + (1 + \beta)p_2(i)] \end{cases} \quad (2)$$

with a spread factor β defined by (3),

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta+1}} & \text{if } u < 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases} \quad (3)$$

where u is a random variable in the interval $[0,1]$ and η is a nonnegative real number that characterizes the distribution of the children in relation to their parents. A large value of η gives a higher probability for creating children near parents. Acting alone and without any mutation operator, SBX presents interesting properties of self-adaptation similarly to Evolution Strategies [9].

3.3 The Breeder Genetic Crossover

From two parent solutions $p_1(i)$ and $p_2(i)$, the BGX crossover creates one child $c(i)$ as follows [10]:

$$c(i) = p_1(i) \pm \frac{(p_2(i) - p_1(i))}{\|p_2(i) - p_1(i)\|} \Delta_i \delta \quad (4)$$

where Δ_i is normally set to 0.5 times the domain of definition of the object variable i and the metric denotes the Euclidean distance in the object variable space. δ is computed from a distribution that favors small values:

$$\delta = 2^{-ku} \quad (5)$$

where u is a random variable in the interval $[0,1]$, the precision constant k being typically set to 16. Note that in [10], the child was placed more often in the direction to the best parent ($p_1(i)$ being the parent with the better fitness) and the minus sign in (4) was chosen with probability 0.9. In this work, the choice of $p_1(i)$ and the sign in (4) are made with a probability 0.5.

In Fig. 1, the probability density function per child for the previous investigated crossover operators is depicted. The corresponding parents p_1 and p_2 are marked with a full circle. Note these three crossover operators are rather complementary since BGX with $k=16$ essentially reinforces the accuracy in the neighborhood of the parents whereas SBX-0.5 favors global exploration.

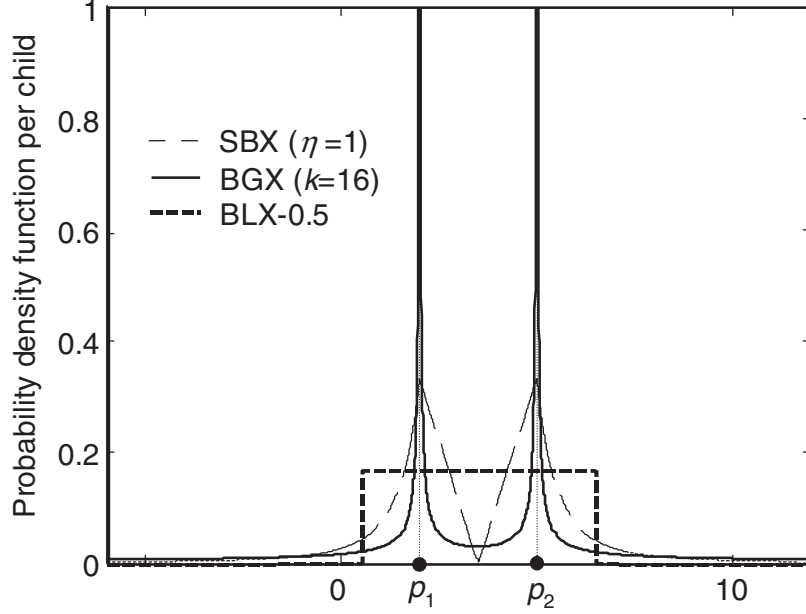


Fig. 1. Probability distribution of children solutions related to each crossover operators

3.4 Self-Adaptive Mutation

We also investigate a self-adaptive mutation operator used in standard Evolution Strategies [11], [9]. Each individual is characterized by its object variables $x(i)$ and associated standard deviation $\sigma(i)$. Children $(c(i), \sigma_c(i))$ are created from their parents $(p(i), \sigma_p(i))$ using the following rule :

$$\sigma_c(i) = \sigma_p(i) \exp(\tau' N(0,1) + \tau N_i(0,1)) \quad (6)$$

$$c(i) = p(i) + \sigma_c(i) N_i(0,1) \quad (7)$$

where $N(0,1)$ denotes a normally distributed random number with mean 0 and standard deviation 1. $N_i(0,1)$ indicates that the random number is generated anew for each value of i . The factors τ and τ' are commonly set to $(2m^{1/2})^{-1/2}$ and $(2m)^{-1/2}$ where m denotes the number of object variables [11].

3.5 Self-Adaptive Recombination

As it is not possible to *a priori* know which crossover operator will be the most efficient on a specific problem, we propose a self-adaptive scheme similar to that of Spears for binary encoded GAs [12]. It consists in associating in the chromosome of individuals an additional gene (*X-gene*) that codes the type of crossover to apply during the recombination. When recombining two parents, the operating crossover is randomly chosen from the *X-gene* of the parents. Using this procedure, the MOGA will favor the crossover that produces the best children through the selection operator. To avoid premature convergence to a particular type of crossover, the *X-gene* also undergoes mutation.

4 Experimental Tests

4.1 Test Problems

We consider three multi-objective problems of the literature [13], [14] displayed in Table 1. ZDT4 is a multimodal continuous problem, which contains 21^9 local Pareto fronts. The global Pareto front is obtained with $g=1$ and is convex. ZTD6 has a non-uniformly distributed search space with solutions non-uniformly distributed along the Pareto front (the front is biased for solutions for which $f_1(x_1)$ is close to one). The Pareto front is obtained with $g=1$ and is non-convex. SCH is a generalization of the Schaffer's problem. A large variable space domain and a convex Pareto front characterize it.

Table 1. Test problems used in this study (minimization of both objectives)

Problem	Characteristics
ZDT4	$f_1(x) = x_1 \quad 0 \leq x_1 \leq 1$ $f_2(x) = g \left(1 - \sqrt{\frac{x_1}{g}} \right) \quad -5 \leq x_i \leq 5 \quad i = 2, \dots, 10$ <p>where $g = 91 + \sum_{i=2}^{10} (x_i^2 - 10 \cos(4\pi x_i))$</p>
ZDT6	$f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(x) = g(1 - (f_1/g)^2) \quad 0 \leq x_i \leq 1 \quad i = 1, \dots, 10$ <p>where $g = 1 + 9 \left(\sum_{i=2}^{10} x_i / 9 \right)^{0.25}$</p>
SCH	$f_1(x) = \frac{1}{40} \sum_{i=1}^{10} x_i^2 \quad -1000 \leq x_i \leq 1000 \quad i = 1, \dots, 10$ $f_2(x) = \frac{1}{40} \sum_{i=1}^{10} (x_i - 2)^2$

4.2 Performance Criteria

Three performance criteria are used to assess the efficiency of MOGAs :

Average deviation to the theoretical Pareto-optimal front. The average distance $\bar{\epsilon}$ of the non-dominated set to the Pareto-optimal front is computed as follows [13],

$$\bar{\epsilon} = \frac{1}{|F|} \sum_{a \in F} \min \{ \| a - a^* \| \mid a^* \in F^* \} \quad (8)$$

where F (respectively F^*) denotes the non-dominated set in the final population (respectively the theoretical Pareto-optimal front), a and a^* belonging to each subset. The metric in (8) is the Euclidean distance computed in the objective space.

Spread. We define the spread $\bar{\varepsilon}_{\min}$ as the average minimum distance of the non-dominated set to the Pareto-optimal solutions that minimizes each objective independently.

$$\bar{\varepsilon}_{\min} = \frac{1}{n} \sum_{i=1}^n \min \left\{ \| a - a_{i\min}^* \| \mid a \in F \right\} \quad (9)$$

where n is the number of objectives and $a_{i\min}^*$ represents the theoretical solution of the Pareto-optimal front that minimizes the i -th objective. Spread characterizes the ability of MOGAs to detect boundary solutions of the Pareto-optimal front.

Spacing. Spacing Δ is a measure based on consecutive distances among the solutions of the non-dominated set [3]. It assesses the ability of the MOGAs to distribute its population uniformly along the Pareto-optimal front.

$$\Delta = \frac{1}{|F|-1} \sum_{i=1}^{|F|-1} |d_i - \bar{d}| \quad (10)$$

where d_i is the Euclidean distance between two consecutive solutions of the non-dominated set, \bar{d} being the average of these distances. A value of zero for this metric indicates all the non-dominated solutions found are equidistantly spaced. Unlike the definition of Δ in [3], we do not include in the non-dominated set the boundary solutions of the theoretical Pareto-optimal front to take into account the spread (spread is independently evaluated by (9)).

Note that these criteria are complementary to assess the efficiency of MOGAs as indicated by Fig. 2 which illustrates various situations of the non-dominated set in relation to the theoretical Pareto-optimal front.

5 Tests Results

We successively compare the efficiency of NSGA-II and SPEA2 on the previous test problems using each crossover operators and self-adaptive procedures of sect. 3. All tests are made with the same number of objective function evaluations. NSGA-II and SPEA2 are run for 200 generations with a population size of 100. The archive size is also set to 100 and the crossover probability is 1. Both MOGAs use the BGA mutation operator [10] with a mutation rate of $1/m$ (where m is the number of variables). The self-adaptive recombination scheme employs the BGX with $k=16$, the SBX with $\eta=1$ and the BLX-0.5. The *X-gene* undergoes mutation with a probability of 5%. NSGA-II and SPEA2 with self-adaptive mutations operate with initial standard deviations set to 1/10 times the domain of definition of the object variables and without

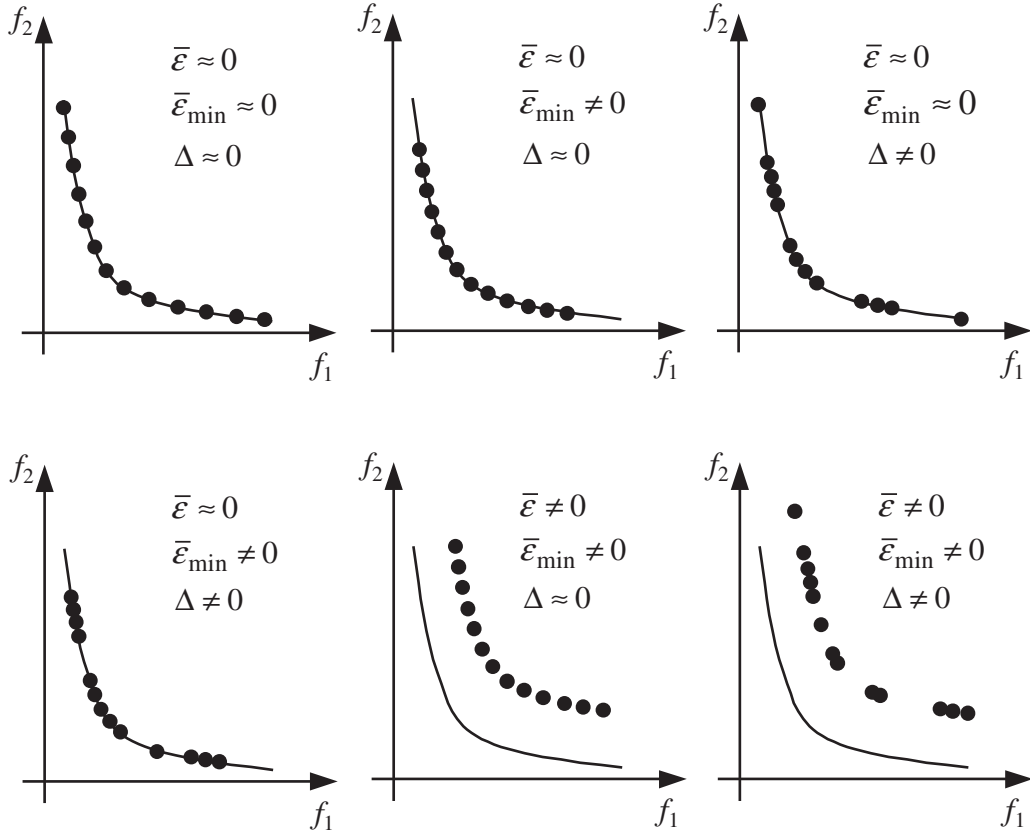


Fig. 2. Illustration of the performance criteria used in this study on various situations. The non-dominated set is symbolized with full circles and the theoretical Pareto optimal front is represented by a continuous line. $\bar{\epsilon} \approx 0$ (resp. $\bar{\epsilon} \neq 0$) indicates a good (resp. a bad) average deviation, $\bar{\epsilon}_{\min} \approx 0$ (resp. $\bar{\epsilon}_{\min} \neq 0$) a good (resp. a bad) spread and $\Delta \approx 0$ (resp. $\Delta \neq 0$) a good (resp. a bad) spacing

any crossover operator. For all investigated tests, 100 runs are made with random populations to take into account the stochastic nature of the MOGA. An average statistic is taken from the final population for the performance criteria.

5.1 Influence of the Recombination Operators

We present in Tables 2–4 the values of the performance criteria on the investigated problems for the NSGA-II and the SPEA2 in relation to each recombination and self-adaptive procedures. From these results, we propose a sort of the recombination and self-adaptive operators on these test problems for each MOGA. Note that this sort was established by giving priority to the average deviation $\bar{\epsilon}$.

As also underlined in a earlier study [2], it can be seen that MOGAs without crossover operator perform poorly even if they use self-adaptive mutations. Table 2 shows that NSGA-II and SPEA2 with BLX-0.5 give the best results on ZDT4. BLX-0.5 associated with the SPEA2 works well on ZDT6 but performs poorly when it is coupled with the NSGA-II. This behavior will be also noted for the self-adaptive recombination scheme in the following section. NSGA-II with the BGX crossover

works well on ZDT6 but performs extremely poorly on SCH (convergence was not achieved after 200 generations; only one non-dominated individual was found in the final population in all runs). Except with the SBX crossover, NSGA-II and SPEA2 fail to spread correctly their population on SCH. Therefore, SBX is ranked at the top of the sort for this problem despite a slightly lowest quality for $\bar{\epsilon}$ and Δ .

Finally, it can be seen that *best results are always obtained by simple crossover acting alone*. However, the efficiency of each crossover operator clearly depends on the characteristics of the test problem. Therefore, using simultaneously multiple crossover operators through a self-adaptive recombination scheme tends to improve the robustness of the MOGAs. We verify this property since the self-adaptive scheme performs extremely well in all cases (the ranking efficiency always equals 2).

Table 2. Performance criteria on problem ZDT4. Results are averaged on 100 runs with random initial populations. Best results are indicated in bold types and margin errors with 95% confidence are given in brackets

MOGA	Operator	Av. dev. $\bar{\epsilon}$	Spread $\bar{\epsilon}_{\min}$	Spacing Δ	Ranking
NSGA-II	BGX ($k=16$)	2.231 [0.185]	1.769 [0.148]	0.112 [0.011]	4 (poor)
	SBX ($\eta=1$)	1.961 [0.170]	1.656 [0.138]	0.033 [0.009]	3 (good)
	BLX-0.5	1.483 [0.151]	1.275 [0.122]	0.019 [0.006]	1 (excellent)
	Self-Ad. Mut.	10.41 [1.000]	9.439 [0.939]	0.017 [0.001]	5 (very poor)
	Self-Ad. Rec.	1.688 [0.164]	1.437 [0.136]	0.021 [0.008]	2 (very good)
SPEA2	BGX ($k=16$)	5.343 [0.436]	4.367 [0.380]	0.301 [0.041]	4 (poor)
	SBX ($\eta=1$)	2.233 [0.221]	1.894 [0.163]	0.028 [0.008]	3 (good)
	BLX-0.5	1.527 [0.148]	1.373 [0.122]	0.021 [0.005]	1 (excellent)
	Self-Ad. Mut.	10.55 [1.090]	9.525 [1.015]	0.012 [0.002]	5 (very poor)
	Self-Ad. Rec.	1.616 [0.161]	1.427 [0.133]	0.024 [0.006]	2 (very good)

Table 3. Performance criteria on problem ZDT6. Results are averaged on 100 runs with random initial populations. Best results are indicated in bold types and margin errors with 95% confidence are given in brackets

MOGA	Operator	Av. dev. $\bar{\epsilon}$	Spread $\bar{\epsilon}_{\min}$	Spacing Δ	Ranking
NSGA-II	BGX ($k=16$)	0.000 [0.000]	0.000 [0.000]	0.006 [0.000]	1 (excellent)
	SBX ($\eta=1$)	0.185 [0.018]	0.001 [0.000]	0.202 [0.022]	4 (poor)
	BLX-0.5	0.081 [0.044]	0.000 [0.000]	0.096 [0.054]	3 (good)
	Self-Ad. Mut.	1.866 [0.114]	0.442 [0.077]	0.543 [0.063]	5 (very poor)
	Self-Ad. Rec.	0.014 [0.006]	0.000 [0.000]	0.024 [0.010]	2 (very good)
SPEA2	BGX ($k=16$)	0.068 [0.008]	0.000 [0.000]	0.071 [0.006]	3 (good)
	SBX ($\eta=1$)	0.186 [0.016]	0.000 [0.000]	0.141 [0.015]	4 (poor)
	BLX-0.5	0.059 [0.005]	0.000 [0.000]	0.055 [0.005]	1 (very good)
	Self-Ad. Mut.	1.662 [0.126]	0.554 [0.090]	0.484 [0.049]	5 (very poor)
	Self-Ad. Rec.	0.061 [0.006]	0.000 [0.000]	0.056 [0.007]	2 (very good)

Table 4. Performance criteria on problem SCH. Results are averaged on 100 runs with random initial populations. Best results are indicated in bold types and margin errors with 95% confidence are given in brackets

MOGA	Operator	Av. dev. $\bar{\epsilon}$	Spread $\bar{\epsilon}_{\min}$	Spacing Δ	Ranking
NSGA-II	BGX ($k=16$)	no convergence achieved in 200 generations			5 (very poor)
	SBX ($\eta=1$)	0.007 [0.000]	0.086 [0.006]	0.006 [0.000]	1 (excellent)
	BLX-0.5	0.004 [0.000]	0.209 [0.008]	0.004 [0.000]	3 (good)
	Self-Ad. Mut.	0.091 [0.006]	0.195 [0.059]	0.008 [0.001]	4 (poor)
	Self-Ad. Rec.	0.004 [0.000]	0.118 [0.009]	0.005 [0.000]	2 (very good)
SPEA2	BGX ($k=16$)	no convergence achieved in 200 generations			5 (very poor)
	SBX ($\eta=1$)	0.006 [0.000]	0.100 [0.007]	0.005 [0.000]	1 (very good)
	BLX-0.5	0.009 [0.001]	0.289 [0.010]	0.002 [0.000]	3 (good)
	Self-Ad. Mut.	0.302 [0.306]	0.445 [0.299]	0.008 [0.002]	4 (poor)
	Self-Ad. Rec.	0.006 [0.000]	0.129 [0.009]	0.004 [0.000]	2 (very good)

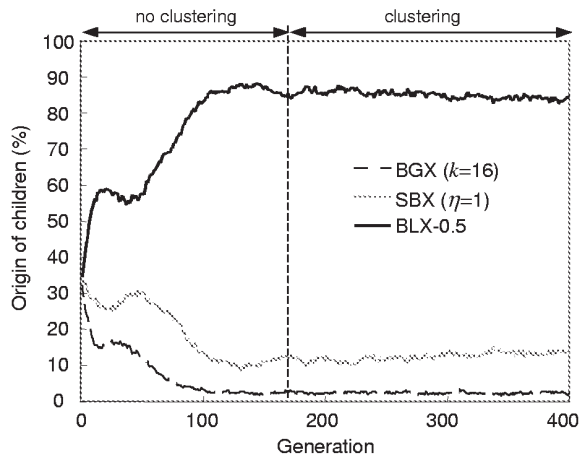
5.2 Analysis of the Self-Adaptive Recombination Scheme

To understand the mechanism of the self-adaptive recombination scheme, we plot in Fig. 3-5 the origin of children for each investigated test problem as a function of the generation number. Results are extended to 400 generations and averaged on 100 runs. We also indicate the threshold from which the clustering is operating.

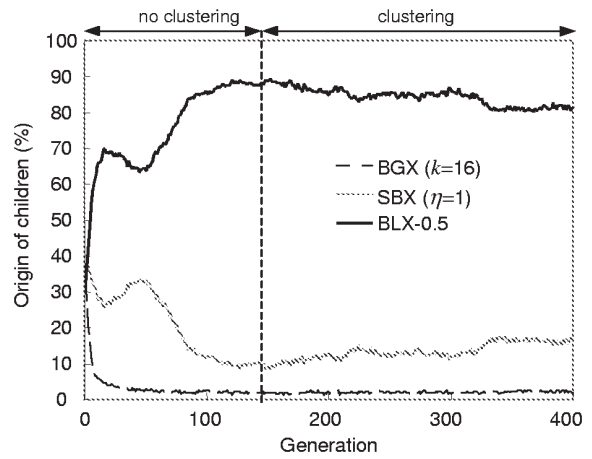
It can be seen that the self-adaptive recombination scheme is able to direct the MOGA towards the crossover operator which performs the best on a given test problem. Therefore, the BLX-0.5 which has the best global exploration properties, is rapidly favored on ZDT4 to avoid misleading local Pareto-optimal fronts (see Fig. 3). Because of the large search space, this crossover is also preferred on SCH at the beginning of the search but both MOGAs finally switch towards the SBX crossover to better distribute individuals on boundary points (see Fig. 5 and Table 3). Curiously, the behavior of the self-adaptive recombination scheme on ZDT6 is radically different for the SPEA2 and the NSGA-II. These results are in accordance with those of sect. 5.1 (see Table 3) which show that SPEA2 performs better with SBX and NSGA-II better with BGX. However, both MOGA need to use the BGX crossover to avoid misleading attractors of high f_2 values in the neighbourhood of the boundary point of the Pareto front defined by $f_1^* \approx 0.28$.

Note that the steady state behavior of the archive characterized by clustering operations does not necessary correspond to the steady state operations of the self-adaptive recombination scheme. Recombination rates of each crossover operator can evolve continuously during the search as shown in Fig 3-5.

Finally, we examine in Table 5-7 the influence of the *X-gene* mutation rate on the efficiency of MOGAs with the studied self-adaptive recombination scheme. Results show that performance criteria are not very sensitive to this factor. However, low mutation rates have to be preferred to exploit benefit of self-adaptation through the selection procedure.

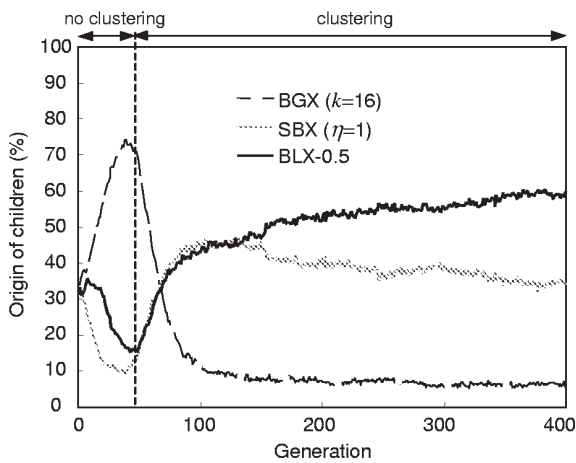


(a) NSGA-II

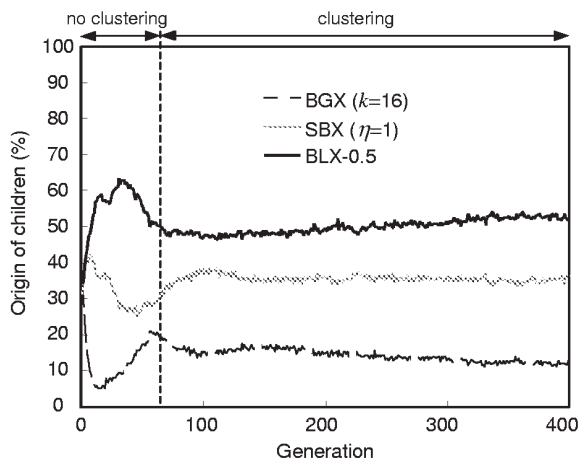


(b) SPEA2

Fig. 3. Origin of children in the self-adaptive recombination scheme. Results on ZDT4 (average on 100 runs)

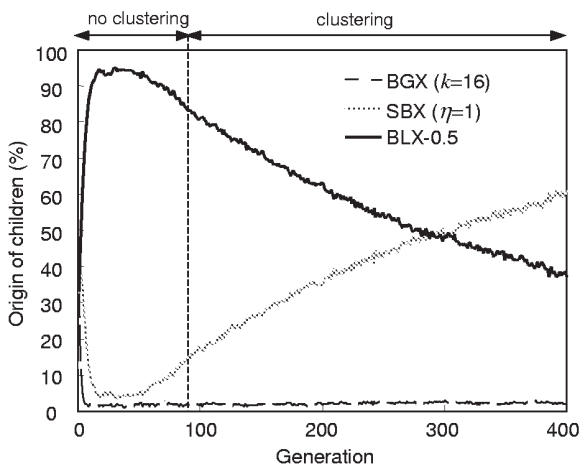


(a) NSGA-II

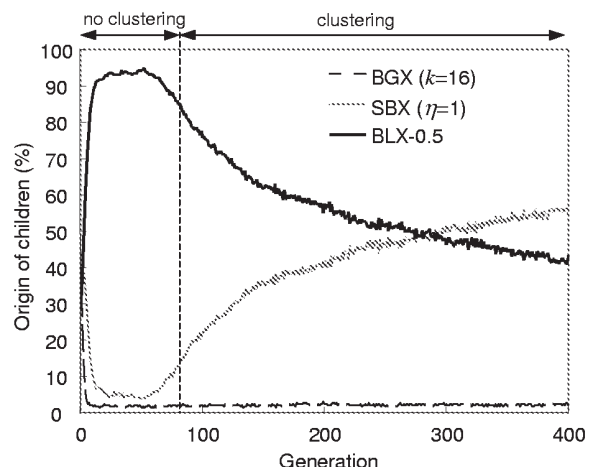


(b) SPEA2

Fig. 4. Origin of children in the self-adaptive recombination scheme. Results on ZDT6 (average on 100 runs)



(a) NSGA-II



(b) SPEA2

Fig. 5. Origin of children in the self-adaptive recombination scheme. Results on SCH (average on 100 runs)

Table 5. Influence of the *X-gene* mutation rate on ZDT4 (average on 100 runs). Best results are indicated in bold types and margin errors with 95% confidence are given in brackets

MOGA	<i>X-gene</i> mutation rate	Av. dev. $\bar{\mathcal{E}}$	Spread $\bar{\mathcal{E}}_{\min}$	Spacing Δ
NSGA-II	0	1.806 [0.198]	1.516 [0.158]	0.027 [0.012]
	5%	1.688 [0.164]	1.437 [0.136]	0.021 [0.008]
	20%	1.699 [0.182]	1.460 [0.148]	0.019 [0.005]
	100% (random case)	2.084 [0.187]	1.783 [0.151]	0.035 [0.010]
SPEA2	0	1.624 [0.138]	1.446 [0.118]	0.018 [0.004]
	5%	1.616 [0.161]	1.427 [0.133]	0.024 [0.006]
	20%	1.733 [0.182]	1.521 [0.135]	0.020 [0.006]
	100% (random case)	2.192 [0.201]	1.838 [0.147]	0.037 [0.009]

Table 6. Influence of the *X-gene* mutation rate on ZDT6 (average on 100 runs). Best results are indicated in bold types and margin errors with 95% confidence are given in brackets

MOGA	<i>X-gene</i> mutation rate	Av. dev. $\bar{\mathcal{E}}$	Spread $\bar{\mathcal{E}}_{\min}$	Spacing Δ
NSGA-II	0	0.005 [0.005]	0.000 [0.000]	0.011 [0.005]
	5%	0.014 [0.006]	0.000 [0.000]	0.024 [0.010]
	20%	0.015 [0.006]	0.000 [0.000]	0.026 [0.010]
	100% (random case)	0.012 [0.004]	0.000 [0.000]	0.023 [0.008]
SPEA2	0	0.076 [0.008]	0.000 [0.000]	0.064 [0.007]
	5%	0.061 [0.006]	0.000 [0.000]	0.056 [0.007]
	20%	0.071 [0.008]	0.000 [0.000]	0.064 [0.008]
	100% (random case)	0.080 [0.007]	0.000 [0.000]	0.069 [0.007]

Table 7. Influence of the *X-gene* mutation rate on SCH (average on 100 runs). Best results are indicated in bold types and margin errors with 95% confidence are given in brackets

MOGA	<i>X-gene</i> mutation rate	Av. dev. $\bar{\mathcal{E}}$	Spread $\bar{\mathcal{E}}_{\min}$	Spacing Δ
NSGA-II	0	0.004 [0.000]	0.210 [0.009]	0.004 [0.000]
	5%	0.004 [0.000]	0.118 [0.009]	0.005 [0.000]
	20%	0.004 [0.000]	0.103 [0.009]	0.005 [0.000]
	100% (random case)	0.006 [0.000]	0.124 [0.009]	0.005 [0.000]
SPEA2	0	0.003 [0.000]	0.289 [0.009]	0.002 [0.000]
	5%	0.006 [0.000]	0.129 [0.009]	0.004 [0.000]
	20%	0.003 [0.000]	0.132 [0.009]	0.004 [0.000]
	100% (random case)	0.006 [0.000]	0.169 [0.009]	0.004 [0.000]

6 Conclusion

In this paper, we have shown that the efficiency of real-encoded MOGAs strongly depends on the crossover operators used to explore new solutions of the search space. Therefore, we have proposed a self-adaptive recombination scheme based on three complementary crossover operators to reduce the sensitivity to the crossover proce-

ture and improve MOGA robustness. First results are promising and further investigations on difficult constrained problems should confirm the advantage of using this technique in MOGAs.

References

1. Corne D.W, Knowles J.D., Oates M.J. : The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lut-ton, J.J. Merelo and H.P. Schwefel eds., Proceedings of the Parallel Problem Solving from Nature VI Conference, Springer (2000) 839–848
2. Costa L., Oliveira P. : An Evolution Strategy for Multiobjective Optimization. Congress on Evolutionary Computation (CEC'2002) IEEE Service Center, Piscataway, New Jersey, Vol. 1 (2002) pp. 97–102
3. Deb K, Agrawal S., Pratab A., Meyarivan T.: A fast-elitist non-dominated sorting genetic algorithm for multiobjective optimization: NSGA-II. Proceeding of the Parallel Problem Solving from Nature VI Conference (2000) 849–858
4. Zitzler E., Laumanns M., Thiele L. : SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty (eds.) EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Ap-plications to Industrial Problems, Athens, Greece, September (2001) 12–21
5. Laumanns M., Rudolph G., Schwefel H.P. : Approximating the Pareto Set: Concepts, Diversity Issues, and Performance Assessment, Technical Report CI-72/99, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany, ISSN 1433-3325 (1999)
6. Laumanns M., Rudolph G., Schwefel H.P. : Mutation Control and Convergence in Evolu-tionary Multi-Objective Optimization. In R. Matousek and P. Osmera (eds.), Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001), Brno University of Technology, Brno, Czech Republic, (2001) 97–106
7. Eshelman L.J., Schaffer J.D. : Real-coded genetic algorithms and interval schemata. In D. Whitley (Ed). Foundations of Genetic Algorithms II (1993) 187–202
8. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
9. Deb K, Beyer H.G : Self-Adaptation in Real-Parameter Genetic Algorithms with Simu-lated Binary Crossover. Genetic and Evolutionary Computation Conference (GECCO-99), Orlando, FL. (1999)
10. Schlierkamp-Voosen D., Mühlenbein H. : Strategy Adaptation by Competing Subpopula-tions. Parallel Problem Solving from Nature 3 – PPSN III, Jerusalem, Springer (1994) 199–208
11. Bäck Th.: Evolutionary algorithms in Theory and Practice. Oxford University Press, New York (1996)
12. Spears W.M. : Adapting crossover in evolutionary algorithms. Proceeding of the 5th An-nual Conference on Evolutionary Programming, San Diego, CA, Morgan Kaufmann Pub-lishers (1995)
13. Zitzler E., Deb K., Thiele L. : Comparison of multiobjective evolutionary algorithms : Empirical results. Evolutionary Computation 8 (2) (2000) 173–195
14. Schaffer J.D. : Multiple objective optimization with vector evaluated genetic algorithms. In J.J. Grefenstette ed. Proceedings of the First International Conference on Genetic Algo-rithms and Their Applications, Pittsburgh, PA (1985) 93–100