Analysis and Enhancement of Simulated Binary Crossover

Joel Chacón, Carlos Segura
Center for Research in Mathematics, Jalisco S/N, Col. Valenciana CP: 36023 Guanajuato, Gto, México
joel.chacon@cimat.mx, carlos.segura@cimat.mx

***Abstract*—Most recombination operators are designed with the aim of altering its exploration capabilities depending on the distance between the parents involved in the process. However, relating the exploration capability of crossover operators only to the distance between parents might be a drawback in long-term executions because diversity could not be large enough after some generations, so the search process might stagnate. This paper proposes some extensions of the Simulated Binary Crossover (SBX) to generate dynamic variants of SBX (DSBX). The dynamic variants consider the stopping criterion to alter their internal operation. The main objective of the extensions is to induce a gradual change from exploration to intensification in the search process, which is performed by dynamically altering three different features of the original SBX. In order to validate the effectiveness of our proposal DSBX is integrated in three different state-of-the-art Multi-objective Evolutionary Algorithms (MOEAs). The experimental validation performed with the popular DTLZ, WFG and UF benchmarks shows a significant improvement of all the MOEAs when applying the novel crossover operator. Additionally, our proposal is also tested against schemes that incorporate differential evolution operators, showing quite competitive results.**

I. Introduction

Evolutionary Algorithms (EAs) are one of the most promising alternatives for dealing with optimization problems when exact approaches are not suitable. Multi-objective optimization is one of the fields where EAs have gained more popularity in the last decades. Multi-objective Optimization Problems (MOPs) involve the simultaneous optimization of two or more objective functions that are usually in conflict. Particularly, a continuous minimization MOP, which is the kind of problems taken into account in this paper, are defined as follows:

$$\begin{array}{ll} minimize & F(x) = (f_1(x), f_2(x), ..., f_m(x)) \\ subject \quad to & x \in \Omega \end{array} \tag{1}$$

where $x = (x_1, ..., x_n) \in R^n$ is the decision variable vector, n corresponds to the number of decision variables, Ω is the feasible space, $F : \Omega \rightarrow R^m$ consists of m objective functions and R^m is known as the *objective space*.

In a minimization MOP with m objective functions, and given two solutions x, $y \in \Omega$, x dominates y, denoted by $x \prec y$, if $f_i(x) \leq f_i(y)$ for all objectives $\{1, ..., m\}$, and $F(x) \neq F(y)$. This means that solution x is not worse than y in any of the objectives and x is strictly better than y in at least one objective. In terms of the Pareto dominance definition, the best solutions of a MOP are those whose objective vectors are not dominated by any other feasible vector. These kinds of solutions are known as Pareto optimal solutions. The Pareto set is the set of all the Pareto optimal solutions whereas the Pareto front are the images of the Pareto set. The goal of multi-objective optimization approaches is to obtain a proper approximation of the Pareto front. Particularly, a set of solutions that are diverse and close to the Pareto front are desired.

In the last decades several categories of Multi-Objective Evolutionary Algorithms (MOEAs) have appeared [1], [2]. Among them, the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [3], the MOEA based on decomposition (MOEA/D) [4], and the S-metric Selection Evolutionary Multi-objective Optimization Algorithm (SMS-EMOA) [5] are proper representatives of three different ways of designing MOEAs. The reproduction phase of these methods is usually based on the use of classical mutation and crossover operators, such as the Simulated Binary Crossover (SBX) and polynomial mutation. However, some methods that are not based on traditional operators, such as Differential Evolution (DE), have gained a lot of popularity in the last years and in many cases have reported better results than methods with classical operators [6], [7]. In this sense, the DE operators are very promising. However, it is well known that several components of DE induce a fast convergence because some greedy decisions are performed which might be counterproductive in long-term executions [8]. Thus, while inducing a fast convergence might be appropriate in short-term executions, it might not be adequate in other cases. As a result, the velocity of convergence should depend on the stopping criterion to successfully profit from the available computing resources. In order to deal with these drawbacks, some of the most recent algorithms consider the stopping criterion to alter their internal behavior. This is the case of some methods with adaptive parameters [9], variable population size [10] and/or novel mutation strategies [11]. As an alternative, more traditional EAs that include operators based on the use of density distributions to generate offspring usually induce a slower convergence meaning that they generate solutions with a lower quality than DE in short-term executions both in continuous and discrete domains [6], [7]. Nevertheless, since they maintain a larger diversity and use operators that usually induce a more explorative behavior, they might be more appropriate for long-term executions.

The focus of this paper is to provide a novel recombination operator that includes the principle of adapting its behavior depending on the stopping criterion set in the EA. The main objective is to enhance the search capability of EAs by inducing a gradual change between exploration to intensification in the search process. In order to design the novel operator, the well known SBX is analyzed and

978-1-5090-6017-7/18/$31.00 ©2018 IEEE

extended. Particularly, three different components of SBX that are suitable to be altered dynamically are taken into account. The novel operator is integrated with state-of-the-art MOEAs and compared with more traditional operators and with variants that consider the DE operators in several well known benchmarks. Results confirm the superiority of the proposal both in terms of hypervolume and inverted generational distance (IGD+).

The rest of this paper is organized as follows. A brief description of the state-of-the-art MOEAs and a detailed revision of the SBX operator is presented in section II. This section includes some empirical studies of the SBX operator to better justify our proposal. Based on these studies, a dynamic variant of the SBX is proposed in section III. The experimental validation is given in Section IV. Finally, conclusions and some lines of future work are outlined in Section V.

II. Literature Review

This section is devoted to review some of the most important works that are highly related to the research presented in this paper. First, the most important MOEAs paradigms are defined. Thereafter, some relevant classifications of crossover operators are introduced. Finally, the popular SBX operator, which is used extensively in this paper, is discussed.

A. Multi-objective Evolutionary Algorithms

In the last years, a large number of MOEAs following different design principles have been devised. In order to better classify them, several taxonomies have been proposed [12]. Attending to the principles of design, MOEAs can be based on Pareto dominance, indicators and/or decomposition [13]. All of them have quite competitive representatives, so in this paper MOEAs belonging to the different groups are taken into account. Particularly, the experimental validation has been carried out by including the Non-Dominated Sorting Genetic Algorithm (NSGA-II) [3], the MOEA based on Decomposition (MOEA/D) [4], and the S-Metric Selection Evolutionary Multi-objective Optimization Algorithm (SMS-EMOA) [5]. They are representative methods of the domination-based, decomposition-based and indicator-based paradigms, respectively. The following subsections briefly describe each one of these paradigms and introduce the selected methods.

1) Domination Based MOEAs - NSGA-II: One of the most recognized paradigms is the domination based approach. MOEAs belonging to this category are based on the application of the dominance relation to design different components of the EAs, specially the selection phase. Given that the dominance relation does not inherently promotes diversity in the objective space, auxiliary techniques such as niching, crowding and/or clustering are usually integrated to obtain an acceptable spread and diversity in the objective space. A critical drawback of methods based on the dominance relation is its scalability in terms of the dimensionality of the objective space. In fact the selection pressure is substantially reduced as the number of objectives increases. Although some strategies have been developed to deal with this issue [14] it remains as an important drawback for this kind of algorithms.

One of the most popular techniques of this group is the NSGA-II. This algorithm [3] considers a special selection operator based on non-dominated sorting and crowding. Non-dominated sorting is used to provide convergence to the Pareto front whereas crowding promotes the preservation of diversity in the objective space.

2) Decomposition Based MOEAs - MOEA/D: Decomposition-based MOEAs [4] transform a MOP in a set of single-objective optimization problems that are tackled simultaneously. This transformation can be achieved through several approaches. The most popular of them is applying a weighted Tchebycheff function, thus requiring a set of well distributed weights to attain well-spread solutions. An important drawback of this kind of approaches is related to the dependency between the Pareto front geometry and the weights required to attain proper solutions.

MOEA/D [4] is a recently designed decomposition-based MOEA. It includes several features such as problem decomposition, weighted aggregation of objectives and mating restrictions based on neighborhood definitions. Particularly, the neighborhoods are considered in the mating selection. A popular variant of MOEA/D is the MOEA/D-DE, which uses the DE operators [15] and the polynomial mutation operator [16] in the reproduction phase. Additionally, it has two extra mechanisms for maintaining the population diversity [17].

3) Indicator Based MOEAs - SMS-EMOA: In multi-objective optimization several quality indicators have been developed to compare the performance of MOEAs. Since these indicators measure the quality of the approximations attained by MOEAs, a paradigm based on the application of these indicators was proposed. Particularly, the indicators replace the Pareto dominance relation with the aim of guiding the optimization process. Among the different indicators, hypervolume is a widely accepted Pareto-compliance quality indicator [18]. One of the main advantages of these algorithms is that indicators usually take into account both the quality and diversity of the solutions, so no additional mechanisms to preserve diversity are required.

A popular and extensively used indicator-based algorithm is the SMS-EMOA [5]. This algorithm might be considered as hybrid, since it involves both indicators and Pareto dominance concepts. Essentially, it integrates the non-dominated sorting method with the use of the hypervolume metric. Thus, SMS-EMOA uses the hypervolume as a density estimator which results in a computationally expensive task. Particularly, the replacement phase erases the individual of the worst ranked front with the minimum contribution to the hypervolume. Taking into account the promising behavior of SMS-EMOA, it has been used in our experimental validation.

B. Crossover operators

The crossover operators are designed to generate offspring solutions using information of the parent solutions. They combine features of two or more parent solutions to generate

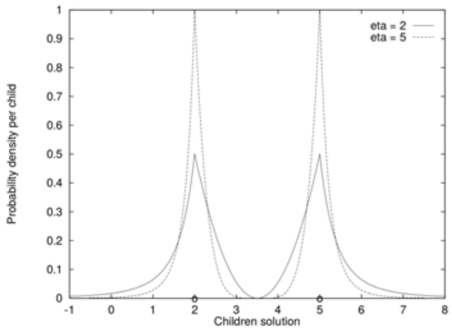

Fig. 1. Probability density function of the SBX operator with indexes of distribution 2 and 5. The parents are located in 2 and 5 respectively.

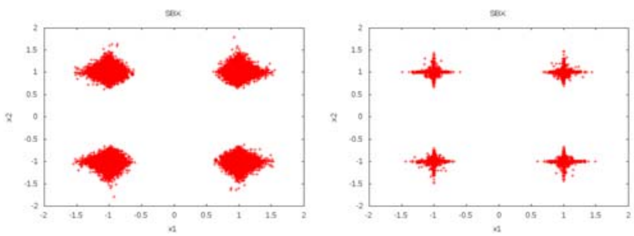

Fig. 2. Simulations of the SBX operator with a distribution index equal to 20. Parents are located in $P_1 = (-1.0, -1.0)$ and $P_2 = (1.0, 1.0)$. The left simulation corresponds to a probability of altering a variable (δ_1 in Algorithm 1) equal to 1.0 and in the right it corresponds to 0.1.

new candidate solutions. Since several crossover operators have been proposed, some taxonomies have also been provided. The taxonomies are based on features such as the location of new generated solutions or the kinds of relations among the variables.

A popular taxonomy classifies crossover operators into variable-wise operators and vector-wise operators. In the variable wise category, each variable from parent solutions is recombined independently with a certain pre-specified probability to create new values. These operators are specially suitable to deal with separable problems. Some operators belonging to this category are the Blend Crossover (BLX) [19], and the SBX [20]. Alternatively, the vector-wise recombination operators are designed to take into account the linkage among variables. They usually perform a linear combination of the variable vectors. Some operators belonging to this category are the Unimodal Normally Distributed Crossover (UNDX) [21], and the simplex crossover (SPX) [22]. Additionally, crossover operators can be classified as Parent-Centric and Mean-Centric [23]. In Parent-Centric operators, children solutions are created around one of the parent solutions, whereas in Mean-Centric operators, children solutions tend to be created mostly around the mean of the participating parent solutions. Among the crossover operators, SBX is probably the most frequently used operator, so this research focuses on this crossover.

1) Simulated Binary Crossover - SBX: The reproduction operators are one of the most relevant components that influence the search process of EAs. Specifically, the crossover

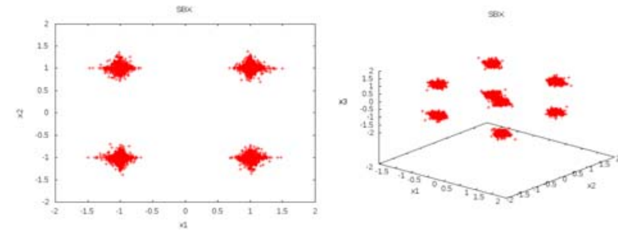

Fig. 3. Simulations of the SBX operator with a distribution index equal to 20. Parents are located in $P_1 = (-1.0, -1.0)$ and $P_2 = (1.0, 1.0)$ and $P_1 = (-1.0, -1.0, -1.0)$ and $P_2 = (1.0, 1.0, 1.0)$ for two and three variables respectively.

and mutation operators are highly related with the diversity of solutions. Hence, the quality of solutions are highly affected by the applied operators.

Simulated Binary Crossover (SBX) [24] is probably the most popular operator for continuous domains and most MOEAs have been extensively tested with such an operator [3], [5]. SBX is classified as Parent-Centric, meaning that two children values (c_1 and c_2) are created around the parent values (p_1 and p_2). The process of generating the children values is based on a probability distribution. This distribution controls the spread factor $\beta = |c_1 - c_2|/|p_1 - p_2|$ defined as the ratio between the spread of the children values and parent values. In order to define this density function a distribution index η_c (a user-defined control parameter) alters the exploration capability of the operator. Specifically, a small index induces a larger probability of building children values distant to the parent values, whereas high indexes tend to create solutions very similar to the parents as is shown in Figure 1.

The probability distribution to create an offspring value is defined as a function of $\beta \in [0, \infty]$ as follows:

$$P(\beta) = \begin{cases} 0.5(\eta_c + 1)\beta^{\eta_c}, & \text{if} \quad \beta \leq 1 \\ 0.5(\eta_c + 1)\frac{1}{\beta^{\eta_c + 2}}, & \text{otherwise} \end{cases} \tag{2}$$

Based in the mean-preserving property of children values and parent values, SBX has the following properties:

- Both offspring values are equi-distant from parent values.
- There exist a non-zero probability to create offspring solutions in the entire feasible space from any two parent values.
- The overall probability of creating a pair of offspring values within the range of parent values is identical to the overall probability of creating two offspring values outside the range of parent values.

Therefore, considering two participating parent values (p_1 and p_2), two offspring values (c_1 and c_2) can be created as linear combination of parent values with a uniform random number $u \in [0, 1]$, as follows:

$$\begin{aligned} c_1 &= 0.5(1 + \beta(u))p_1 + 0.5(1 - \beta(u))p_2 \\ c_2 &= 0.5(1 - \beta(u))p_1 + 0.5(1 + \beta(u))p_2 \end{aligned} \tag{3}$$

Algorithm 1 Simulated Binary Crossover (SBX)

Input: Parents (P_1, P_2), Distribution index (η_c), Crossover probability (P_c).
Output: Children (C_1, C_2).
if $U[0,1] \leq P_c$ **then**
  for each variable d **do**
    if $U[0,1] \leq \delta_1$ **then**
      Generate $C_{1,d}$ with Equations (5) and (6).
      Generate $C_{2,d}$ with Equations (5) and (7).
      if $U[0,1] \leq (1-\delta_2)$ **then**
        Swap $C_{1,d}$ with $C_{2,d}$.
    else
      $C_{1,d} = P_{1,d}$.
      $C_{2,d} = P_{2,d}$.
else
  $C_1 = P_1$.
  $C_2 = P_2$.

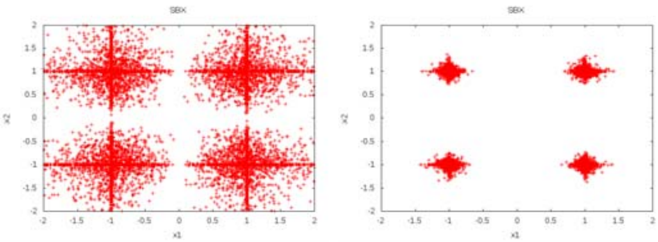

Fig. 4. Simulation of the SBX operator sampling $10,000$ children values, the parents are located in $P_1 = (-1.0, -1.0)$ and $P_2 = (1.0, 1.0)$. The left and right are with a distribution index of 2 and 20 respectively.

The parameter $\beta(u)$ depends on the random number u, as follows:

$$\beta(u) = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if} \quad u \leq 0.5, \\ (\frac{1}{2(1-u)})^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{4}$$

The above equation considers an optimization problem with no variable bounds. In most practical problems, each variable is bounded within a lower and upper bound. Thus, the modification of the probability distribution shown in Equation (5) was proposed [25] with the aim of taking into account such bounds. This last variant is extensively used nowadays.

$$\beta(u) = \begin{cases} (2u(1-\gamma))^{\frac{1}{\eta_c+1}}, & \text{if} \quad u \leq 0.5/(1-\gamma), \\ (\frac{1}{2(1-u(1-\gamma))})^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{5}$$

$$c_1 = 0.5(1+\beta(u))p_1 + 0.5(1-\beta(u))p_2 \tag{6}$$

$$c_2 = 0.5(1+\beta(u))p_1 + 0.5(1-\beta(u))p_2 \tag{7}$$

In this case, the child c_1 which is nearest to p_1 is calculated according to the Equation (6). Considering that $p_1 < p_2$ and with a lower bound equal to a, $\gamma = 1/(\alpha^{\eta_c+1})$, where $\alpha = 1 + (p_1 - a)/(p_2 - p_1)$. Similarly, the second child c_2 is computed with $\alpha = 1 + (b - p_2)/(p_2 - p_1)$, where b correspond to the upper bound. Then, the second child is computed as is indicated in Equation (7).

Note that as reported in [20] several extensions of the SBX to problems with multiple variables might be provided. Authors considered a simple strategy for choosing the variables to cross [21]. Specifically, each variable is crossed with probability 0.5, following the principles of uniform crossover. Authors recognized the important implications on the linkage among variables of such decisions. In any case, this is the most typical way of applying SBX in problems with multiple variables nowadays.

2) Implementation and analyses of SBX operator: This section discusses some of the main characteristics of the most currently used implementation of the SBX operator for problems with multiple variables. Essentially, three key components that might affect its performance are discussed. Firstly, as already mentioned it alters each variable with a fixed probability equal to 0.5. If this probability value is increased, the children tend to be more distant to the parents, since in average more variables are modified simultaneously. In separable problems, altering only one variable might be adequate. However, for non-separable problems altering several variables simultaneously seems more promising. The implications of varying this probability is illustrated in Figure 2, where a problem with two variables is taken into account. In the right side a low probability is used and it provokes a bias to explore by keeping some values intact creating a figure similar to a cross in the two-dimensional case. This feature might be suitable for separable problems. Alternatively, the left side shows that when using a high probability this bias disappears, which could be more suitable for non-separable problems. Note that this probability is somewhat related with the distribution index in the sense that both have a direct effect on the similarity between parents and children.

The second key issue is that after generating the two child values with the SBX distribution, such values are interchanged with a fixed probability that is usually set to 0.5, i.e. the value closer to parent p_1 is not always inherited by c_1. This is a feature that is not usually discussed but it is important for the obtained performance. In some contexts this probability is known as "Variable uniform crossover probability" [6] or "Discrete Recombination" [26]. Since in multi-objective optimization more diversity is maintained these swaps might produce a high disruptive operator. In fact, in some sense due to this action it is not so clear that SBX can be categorized as a parent-centric operator. These interchanges between the children has the effect of performing multiple "reflections" in the search space. When increasing the dimensions of the decision variables the number of regions covered increases exponentially as is illustrated in Figure 3 where cases with two and three decision variables are taken into account. Note also that this feature has a considerable effect on the distance between parents and offspring.

Finally, the last component is the distribution index, which is probably the most well known feature of the SBX. A low index results in greater exploration levels. In fact, a distribution index equal to one has a similar effect to the Fuzzy Recombination Operator [27]. The effect of applying different indexes is illustrated in Figure 4 where the left side considers a low index value whereas the right side takes into account a higher index value, which creates new candidate solutions that are more similar to the parents.

The SBX implementation is shown in Algorithm 1. This pseudocode is based on the implementation that is integrated in the NSGA-II code published by Deb et al. [3] and which is the most popular variant nowadays. As an input it requires two parents (P_1 and P_2) and it creates two children (C_1 and C_2). The first and second key components commented previously correspond to the lines 5 and 8, respectively. As is usual, for the basic case, SBX is configured with $\delta_1 = \delta_2 = 0.5$ and $\eta_c = 20$. It is important take into account that this implementation does not consider the dimension of the decision variables or the stopping criteria to set any of its internal parameters.

III. Proposal

Based on the previous analyses and with the aim of inducing an appropriate balance between exploration and intensification, the following modifications are proposed. First, the probability to modify a variable (δ_1) is dynamically modified during the execution. The rationality behind this modification is to increase the exploration capability in the initial stages by altering simultaneously several variables and then, as the evolution proceeds reduce the number of variables that are modified. The value of δ_1 is changed in base of a linear decreasing model, where initially it is fixed to 1.0 and then it is decreased so that at the half of total generations is equal to 0.5. This last value is maintained until the end of the execution, i.e. from the half of the execution it behaves as the traditional SBX implementation. Equation (8) is the one used to set the value of δ_1, where $G_{Elapsed}$ is the current generation and G_{End} is the total number of generations.

In a similar way, the second change is related to the probability of performing reflections ($1-\delta_2$). In this case δ_2 is also updated as in Equation (8), meaning that the probability of performing a reflection increases from 0.0 to 0.5 during the execution. This modification is performed with the aim of avoiding the disruptive behavior of interchanging the variables at the first generations because this might result in very drastic modifications. Once that the individuals converge to certain degree it might make more sense to perform such reflections. Thus, this probability is increased to 0.5 which is the value used in the standard implementation of SBX.

$$\delta_1 = \delta_2 = max\left(0.5, 1.0 - \frac{G_{Elapsed}}{G_{End}}\right) \quad (8)$$

Finally, the distribution index is also changed during the execution. At the first stages a low distribution index is induced with the aim of increasing the exploration capabilities of SBX. Then, it is linearly incremented which has the effect of closing the distribution curve, meaning that more intensification is promoted. The linear increment is governed by Equation (9), meaning that the distribution index is altered from 2 to 22. Note that modifications similar to this last one have been explored previously [28], [16].

$$\eta_c = 2 + 20 \times \left(\frac{G_{Elapsed}}{G_{End}}\right) \quad (9)$$

TABLE I

References points for the HV indicator

Instances	Reference Point
WFG1-WFG9	$[2.1, \ldots, 2m + 0.1]$
DTLZ 1, 2, 4	$[1.1, \ldots, 1.1]$
DTLZ 3, 5, 6	$[3, \ldots, 3]$
DTLZ7	$[1.1, \ldots, 1.1, 2m]$
UF 1-10	$[2, \ldots, 2]$

IV. Experimental validation

This section is devoted to analyze the results obtained with the dynamic variants of SBX (DSBX). The novel crossover operator was integrated with NSGA-II, MOEA/D and SMS-EMOA. First, three variants that alter only one of each of the components previously discussed are analyzed. Then, a case that alters two of them simultaneously is taken into account. The WFG [29], DTLZ [30] and UF [17] test problems have been used for our purpose. Our experimental validation also includes the variant of Differential Evolution known as DEMO [6] with the aim of comparing our extension of SBX with other well-known operators.

Given that all the methods are stochastic algorithms, each execution was repeated 35 times with different seeds. The common configuration in all of them was the following: the stopping criterion was set to $25,000$ generations, the population size was fixed to 100, WFG test problems were configured with two and three objectives, and 24 variables were considered, where 20 of them are distance parameters and 4 of them are position parameters. In the case of the DTLZ test instances, the number of decision variables were set to $n = M + r - 1$, where $r = \{5, 10, 20\}$ for DTLZ1, DTLZ2 to DTLZ6 and DTLZ7 respectively, as is suggested in [30]. In the UF benchmark set the number of decision variables were set to 10. Finally, the polynomial mutation was used with a mutation probability equal to $1/n$ and with a distribution index equal to 50, whereas for the cases that used the SBX, the crossover probability was set to 0.9 and the distribution index was set to 20. The additional parameterization of each algorithm was as follows:

- **DEMO**: CR = 0.3 and F = 0.5.
- **SMS-EMOA**: offset = 100.
- **MOEA/D**: size of neighborhood = 10, max updates by sub-problem (nr) = 2 and $\delta = 0.9$.

In order to compare the fronts obtained by the different methods the normalized hypervolume (HV) and IGD+ was taken into account. The reference points used for the hypervolume indicator are shown in the Table I and are similar to the ones used in [31], [32].

In order to statistically compare the results (IGD+ and HV values), the following statistical tests were performed. First a Shapiro-Wilk test was performed to check whatever or not the values of the results followed a Gaussian distribution. If, so, the Levene test was used to check for the homogeneity of the variances. If samples had equal variance, an ANOVA test was done; if not, a Welch test was performed. For non-Gaussian distributions, the nonparametric Kruskal-Wallis

TABLE II

STATISTICAL INFORMATION OF METRICS WITH TWO OBJECTIVES

	NSGA-II						MOEA/D						SMS-EMOA					
	1	2	3	4	5	DE	1	2	3	4	5	DE	1	2	3	4	5	DE
Average HV	0.88	0.90	0.90	0.91	0.93	**0.94**	0.87	0.87	0.87	0.90	**0.91**	**0.91**	0.88	0.89	0.87	0.91	0.92	**0.93**
Average IGD+	0.12	0.09	0.11	0.07	0.06	**0.05**	0.14	0.12	0.14	0.09	0.08	**0.07**	0.13	0.11	0.14	0.08	0.07	**0.05**

TABLE III

STATISTICAL INFORMATION OF METRICS WITH THREE OBJECTIVES

	NSGA-II						MOEA/D						SMS-EMOA					
	1	2	3	4	5	DE	1	2	3	4	5	DE	1	2	3	4	5	DE
Average HV	**0.87**	0.84	**0.87**	**0.87**	**0.87**	0.85	0.84	0.84	0.84	**0.86**	**0.86**	0.85	0.90	0.89	0.88	**0.91**	**0.91**	**0.91**
Average IGD+	0.13	0.16	0.13	**0.12**	**0.12**	0.13	0.15	0.14	0.15	**0.11**	**0.11**	0.13	0.11	0.11	0.13	**0.09**	**0.09**	0.13

TABLE IV

SUMMARY OF STATISTICAL TESTS

NSGA-II															
	1			2			3			4			5		
	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$
HV-2obj	16	29	47	6	61	25	28	19	45	31	23	38	**54**	3	35
HV-3obj	15	19	42	12	50	14	17	15	44	**33**	10	33	26	9	41
IGD-2obj	14	30	48	4	60	28	25	17	50	33	19	40	**52**	2	38
IGD-3obj	14	18	44	13	44	19	18	15	43	**33**	15	28	23	9	44

MOEA/D															
	1			2			3			4			5		
	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$
HV-2obj	15	33	44	10	60	22	25	26	41	39	18	35	**57**	9	26
HV-3obj	10	22	44	12	39	25	11	19	46	24	10	42	**38**	5	33
IGD-2obj	16	31	45	9	60	23	23	27	42	37	17	38	**57**	7	28
IGD-3obj	12	22	42	13	43	20	13	24	39	30	9	37	**40**	10	26

SMS-EMOA															
	1			2			3			4			5		
	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$	$\uparrow$	$\downarrow$	$\longleftrightarrow$
HV-2obj	9	35	48	7	43	42	16	31	45	41	9	42	**53**	8	31
HV-3obj	7	21	48	9	35	32	13	21	42	27	6	43	**31**	4	41
IGD-2obj	10	34	48	15	48	29	12	33	47	41	12	39	**55**	6	31
IGD-3obj	8	20	48	13	30	33	9	19	48	22	5	49	**27**	5	44

test was used to test whether samples are drawn from the same distribution. An algorithm X is said to win algorithm Y when the differences between them are statistically significant, and the mean and median obtained by X are higher (in HV) or lower (in IGD+) than the mean and median achieved by Y.

A. Analysis of isolated components

In this section we discuss about the independent effect of each component that is dynamically modified. The effect of each component is analyzed through four cases, based in the Algorithm 1. Each case is described as follows:

- **Case 1**: The standard SBX operator where $\delta_1 = \delta_2 = 0.5$ and $\eta_c = 20$.
- **Case 2**: The value δ_1 is updated according to Equation (8), $\delta_2 = 0.5$ and $\eta_c = 20$.
- **Case 3**: The value δ_2 is updated according to Equation (8), $\delta_1 = 0.5$ and $\eta_c = 20$.
- **Case 4**: The distribution index is updated according to Equation (9), $\delta_1 = \delta_2 = 0.5$.

In order to analyze the performance of each Case (Case 5 is discussed later), Tables II and III shows information about the Normalized Hyper-volume (HV) [28] and about the Inverted Generational Distance Plus (IGD+) [18]. Specifically, the mean of the HV and IGD+ for all considered problems are shown for two and three objectives. It is clear that case 4 outperforms case 1, case 2 and case 3 both with two and three objectives in all the tested algorithms. Therefore, increasing the distribution index during the execution seems to be the most beneficial action. This occurs because the initially open distribution curve leads to a higher degree of exploration, whereas as the evolution proceeds more intensification is promoted. On the other hand, case 2 presented a lower performance than case 1 when taking into account three objectives. Thus, it seems that altering almost all the variables convert the new approach into a too disruptive operator. Perhaps, altering δ_1 in a different way might provide better results, but this is left as a future work.

Previous analyses are only based on the mean obtained for all the problems. However, depending on the problem the performance might vary. This is analyzed in the following section. Additionally, more detailed results are available[1].

[1]https://github.com/joelchaconcastillo/SBX_CEC2018.

B. Simultaneous modification of several components

Based on the previously discussed results, a variant of the SBX is proposed where the case 3 and case 4 are mixed, i.e. both δ_2 and the distribution index are updated dynamically. Since case 2 did not report significant benefits, the updating mechanism for δ_1 was discarded. Specifically in our case 5, Algorithm 1 is configured as follows. The parameter δ_1 is fixed to 0.5, i.e. in a similar way than the standard SBX. Following the case 3, δ_2 is updated according to Equation (8). Finally, according to case 4 η_c is updated in base of Equation (9).

Attending to the mean HV and IGD+ obtained by the case 5 (see Tables II and III) it is clear that integrating case 3 and case 4 is beneficial. The advantages are clearer in the case of two objectives, whereas in the case of three objectives, case 4 and case 5 are similar in terms of mean performance. Moreover, results attained with case 5 are superior to the ones obtained with DE in three objectives, whereas when using the traditional SBX results deteriorate. Thus, when properly configuring a DSBX results similar or superior to DEMO could be obtained.

Finally, since previous analyses only consider the mean among all the benchmark problems, an additional analyses was developed to better understand the contributions of the different cases. Particularly, pair-wise statistical tests among all the five cases that consider SBX and DSBX were carried out. This was performed independently for NSGA-II, MOEA/D and SMS-EMOA. Results of these statistical tests are shown in Table IV. For each algorithm and case, the column "$\uparrow$" reports the number of comparisons where the statistical tests confirmed the superiority of the corresponding case, whereas the column "$\downarrow$" reports the number of cases where it was inferior and "$\longleftrightarrow$" indicates the number of comparisons where differences were not statistically significant. The advantages of case 5 are quite clear. Only in the case of NSGA-II with three objectives, case 4 could outperform the results obtained by case 5. Thus, by properly combining several dynamic modification, results can be improved further. Moreover, results confirm the advantages of our proposals when compared to the standard SBX (case 1). The only case that is not clearly superior to the standard SBX is the case number 2, as it was previously discussed.

V. Conclusions

Crossover is one of the most important operators in EAs. In most cases, crossover operators provide a large degree of exploration when the content of the population is diverse. However, when a low diversity degree is reached, they tend to promote intensification. At the same time, depending on the new candidate solutions created by the crossover operators, the velocity of convergence varies. Since the behavior of operators does not usually depend on the stopping criterion, losing diversity in a gradual way might be a complex task. Thus, some parameterizations might be suitable for some stopping criterion but not for others. This paper proposes extending the well-known SBX by including the stopping criterion and elapsed generations as one of its inputs. First, the standard version of SBX and some actions that can be done to alter its exploration capabilities are analyzed. Specifically, the SBX is composed by three key components that are related with diversity issues. The first controls the amount of variables that are inherited intact. The second is the probability of interchanging new generated values between offspring. Finally, the last component is related with the selected distribution index. These three features are adapted to generate a dynamic SBX (DSBX) by taking into account both the stopping criterion and elapsed generations with the aim of properly altering the degree of exploration and intensification of the novel crossover operator. The experimental validation is carried out with long-term executions and the popular WFG, DTLZ and UF problems. This validation shows that a dynamic distribution index provides significantly better and more robust results than the standard SBX with all the tested MOEAs. Additionally, adapting the probability of interchanging variables between offspring provides benefits and by adapting this probability simultaneously with the distribution index, performance can be further improved. In the case of three objectives, state-of-the-art DE operators could also be outperformed by our proposal. No additional parameterizations adapted to each problem were required to apply the proposed operators, meaning that robust results could be obtained without additional user efforts.

Several lines of the future work might be explored. First, we would like to devise some strategies to adaptively manage the distribution index. Measuring the diversity of the population to alter the behavior of the operators seems a plausible approach. Additionally, using these kinds of operators together with some of the specific methods that has been devised to avoid premature convergence might bring additional benefits. Finally, we plan to use the principles that governed the design of our proposals to devise novel vector-wise operators.

References

[1] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[2] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr 2002.

[4] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, Dec 2007.

[5] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume ," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653 – 1669, 2007.

[6] T. Tušar and B. Filipič, "Differential evolution versus genetic algorithms in multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 257–271.

[7] M. Mi, X. Huifeng, Z. Ming, and G. Yu, "An improved differential evolution algorithm for tsp problem," in *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. 544–547.

[8] Â. A. Sá, A. O. Andrade, A. B. Soares, and S. J. Nasuto, "Exploration vs. exploitation in differential evolution," in *AISB 2008 Convention Communication, Interaction and Social Intelligence*, vol. 1, 2008, p. 57.

[9] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jso," in *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 2017, pp. 1311–1318.
[10] J. Brest and M. S. Maučec, "Population size reduction for the differential evolution algorithm," *Applied Intelligence*, vol. 29, no. 3, pp. 228–247, 2008.
[11] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2011.
[12] A. G. Slim Bechikh, Rituparna Datta, *Recent Advances in Evolutionary Multi-objective Optimization*. springer.
[13] M. Pilát, "Evolutionary multiobjective optimization: A short survey of the state-of-the-art," *Proceedings of the Contributed Papers Part I-Mathematics and Computer Sciences, WDS, Prague, Czech*, pp. 1–4, 2010.
[14] C. Horoba and F. Neumann, "Benefits and drawbacks for the use of epsilon-dominance in evolutionary multi-objective optimization," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 2008, pp. 641–648.
[15] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.
[16] M. Hamdan, "The distribution index in polynomial mutation for evolutionary multiobjective optimisation algorithms: An experimental study," in *International Conference on Electronics Computer Technology, Kanyakumari, India*, 2012.
[17] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of moea/d on cec09 unconstrained mop test instances," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 2009, pp. 203–208.
[18] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 110–125.
[19] L. J. Eshelman, "Real coded genetic algorithms and interval-schemata," *Foundations of genetic algorithms*, vol. 2, pp. 187–202, 1993.
[20] R. B. Agrawal, K. Deb, K. Deb, and R. B. Agrawal, "Simulated binary crossover for continuous search space," Tech. Rep., 1994.
[21] I. Ono, H. Kita, and S. Kobayashi, "Advances in evolutionary computing," A. Ghosh and S. Tsutsui, Eds. New York, NY, USA: Springer-Verlag New York, Inc., 2003, ch. A Real-coded Genetic Algorithm Using the Unimodal Normal Distribution Crossover, pp. 213–237. [Online]. Available: http://dl.acm.org/citation.cfm?id=903758.903767
[22] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *J. of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997. [Online]. Available: http://dx.doi.org/10.1023/A:1008202821328
[23] H. Jain and K. Deb, "Parent to mean-centric self-adaptation in sbx operator for real-parameter optimization," *Swarm, Evolutionary, and Memetic Computing*, pp. 299–306, 2011.
[24] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 3, pp. 1–15, 1994.
[25] K. Deb and H.-G. Beyer, *Self-adaptive genetic algorithms with simulated binary crossover*. Secretary of the SFB 531, 1999.
[26] H. Mühlenbein and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm i. continuous parameter optimization," *Evolutionary computation*, vol. 1, no. 1, pp. 25–49, 1993.
[27] H.-M. Voigt, H. Mühlenbein, and D. Cvetkovic, "Fuzzy recombination for the breeder genetic algorithm," in *Proc. Sixth Int. Conf. on Genetic Algorithms*. Citeseer, 1995.
[28] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
[29] S. Huband, L. Barone, L. While, and P. Hingston, *A Scalable Multi-objective Test Problem Toolkit*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 280–295.
[30] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable test problems for evolutionary multiobjective optimization*. Springer, 2005.
[31] J. A. M. Berenguer and C. A. C. Coello, "Evolutionary many-objective optimization based on kuhn-munkres' algorithm," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 3–17.
[32] Q. Zhu, Q. Lin, Z. Du, Z. Liang, W. Wang, Z. Zhu, J. Chen, P. Huang, and Z. Ming, "A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm ," *Information Sciences*, vol. 345, pp. 177 – 198, 2016.