Parent to Mean-Centric Self-Adaptation in Single and Multi-Objective Real-Parameter Genetic Algorithms with SBX Operator[*]

Kalyanmoy Deb and Himanshu Jain
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
Kanpur, PIN 208016, India
{deb,hjain}@iitk.ac.in
http://www.iitk.ac.in/kangal/deb.htm

KanGAL Report Report Number 2011017

Abstract

Real-parameter optimization using genetic algorithms (GAs) have received significant attention due to their academic value in constrained optimization and also their practical significance. In an earlier study, real-parameter recombination operators were classified into parent-centric or mean-centric categories mainly based on their focus in creating offspring solutions. In this paper, we argue that a self-adaptive determination of parent-centric versus mean-centric approach based on the current population statistics is better than either approach alone. We propose a couple of self-adaptive strategies towards this task and demonstrate the superiority of the approaches over their traditional counterparts on single as well as multi-objective optimization problems. The population approach of GAs facilitates an effective use of the proposed self-adaptive approaches. Further efforts now must be spent to evaluate the approach and suggest other self-adaptive approaches for real-parameter evolutionary optimization.

1 Introduction

One of the challenges in designing an efficient real-parameter evolutionary optimization algorithm is the recombination operator in which two or more population members are blended to create one or more new (offspring) solutions. The existing recombination operators can be classified into two broad categories: (i) variable-wise operators and (ii) vector-wise operators. In the variable-wise category, each variable from participating parent solutions is recombined independently with a certain pre-specified probability to create two new values. The resulting offspring solutions are then formed by concatenating the new values from recombinations or one of the existing parent values, as the case may be. Clearly, this procedure does not honor the necessary *linkage* among different variables that the parent solution have and is a typical difficulty associated with other similar variable-wise recombination operators. These operators are suitable for problems that do not have much linkage among its variables. Some examples of variable-wise recombination operators are blend crossover (BLX) [6], fuzzy recombination (FR) [16], simulated binary crossover (SBX) [2], and others [9]. Despite not honoring linkage explicitly through its operators, these operators are capable of maintaining some linkage information among variables indirectly and have been found to solve many challenging problems. On the other hand, in the vector-wise recombination operators,

[*]Kalyanmoy Deb is also Adjunct Professor of Information and Service Economy, Aalto University School of Economics, Helsinki, Finland.

a linear combination of the complete variable vectors of participating parent solutions is usually made to create an offspring variable vector. Such operators are able to honor the linkage among variables directly. Some examples are UNDX operator [11], simplex crossover (SPX) [15], parent-centric crossover (PCX) [4], and others. It is worth mentioning here that the search operation of CMA-ES [8], correlated evolution strategy [13], differential evolution (DE) [14] and particle swarm optimization (PSO) [10] involve a vector-wise recombination operator.

The recombination operators can also be classified into two functionally different classes based on their emphasis on the location of creating offspring solutions compared to the location of parent solutions. An earlier study [4] classified most of the above-mentioned real-parameter recombination operators into two classes: (i) mean-centric operators and (ii) parent-centric operators. In a mean-centric operator, offspring solutions are created mostly around the mean of the participating parent solutions (variable or vector-wise). BLX, UNDX and SPX operators can be considered to be mean-centric recombination operators. In a parent-centric operator, offspring solutions are created around one of the parent solution (variable or vector-wise). FR, SBX, PCX are examples of such an operator. Since parent solutions are tested to be better by the preceding selection operator (on the contrary, centroid of parents were not tested at all), performing a parent-centric operator was argued to be a better operator than a mean-centric operator. This is particularly true in the beginning of a simulation when parent solutions are away from each other and the centroid of parent solutions is expected to lie completely in a new region which was not adequately tested for their feasibility or objective values.

In this paper, we argue and demonstrate that while a parent-centric operator may be judged to be better early on in a simulation, when population members reach near the optimum of the problem, a mean-centric operator may be found to be more beneficial. Thus, instead of using a parent-centric or a mean-centric recombination operator throughout in a simulation, a better strategy would be to adaptively choose a parent or a mean-centric operator depending on whether the population is approaching or already crowded around the optimum, respectively. Here we also address the issue of respecting the necessary linkage between various variables by implementing the self-adaptive SBX operator variable-wise.

We keep our focus in this paper in addressing two main aspects: (i) whether a self-adaptive recombination approach is better able to search through unimodal and multi-modal objective landscapes than a fixed parent-centric or a mean-centric approach and (ii) if yes, how such a self-adaptation task can be achieved efficiently in a real-parameter recombination operator? To achieve this goal, we do not use all usual variation operators of a RGA, instead confine the algorithm to have the following features: (i) use a generational genetic algorithm, in which the parent population is completely replaced by the offspring population, so that no additional advantage from the generational gap models can be given credit to, (ii) no elite preserving operators used to introduce a non-degrading performance originating from any operation other than the proposed recombination operation, (iii) no mutation operator is used to gain any advantage from another operator in avoiding a premature convergence. Thus, in summary, we use a binary tournament selection to create a mating pool of good solutions from the current population and use the proposed self-adaptive recombination operator alone to create the offspring population. In our opinion, the performance of this simplistic RGA should provide us with an idea of the effectiveness of the proposed recombination operator, particularly when it is compared with a similar RGA with only a difference in its recombination operator. For this unique purpose of this study, we do not compare our selecto-recombinative methodology with any other evolutionary algorithms. Hopefully, the insights obtained in this study in identifying crucial aspects of the working principle of the proposed self-adaptive recombination approaches will be useful in the future to develop a complete evolutionary algorithm and to perform an elaborate comparative study with other competing algorithms.

In the remainder of this paper, first, we overview the difference between parent and mean-centric operators in Section 2 and indicate some popular recombination operators for each type. In Section 3, we argue the importance of both type of operators in an efficient algorithm. We then consider a specific parent-centric operator (SBX) and suggest a modification to the operator so that at every generation it behaves like a parent or a mean-centric operator depending on the population statistics in a self-adaptive manner. The procedure is discussed in detail in Section 4. Thereafter, in Section 5, we demonstrate the efficacy of the proposed recombination operator on a number of unimodal and multi-modal test problems borrowed from the EA literature. The results are compared with the original parent-centric SBX operator and a purely mean-centric recombination operator. In all the problems the proposed methodology outperformed its other competitors. Moreover, the methodology is also found to be scalable from 10 variables to 100 variables. In Section 6, we modify the SA-SBX operator to suggest a variable-wise version that seems to maintain the linkage among the variables. Simulation results using this modified SA-SBX operator are presented next in Section 7. In Section 8, we extend the M-SA-SBX operator to solve multi-objective optimization problems and then presents results on standard two and three-objective test problems. The results are compared with the original NSGA-II procedure. Finally, conclusions of this study is made in Section 9.

2 Real-Parameter Recombination Operators

One of the challenges in designing an efficient real-parameter genetic algorithm (RGA) lies in the creation of its recombination operator [12]. Unlike in the binary-coded GAs, a RGA uses the variables directly as a vector of real-valued numbers. It has been argued and discussed earlier [17, 6, 7] that a straightforward implementation of single or multi-point crossover operation, where the cross-sites are forced to fall in between two variables, is not an efficient approach simply because of the infinite cardinalities involved in the variable values. These studies have led researchers to propose a *blending* operation between two parent values as a viable recombination operator. In this operation, variables are proposed to be recombined one at a time with a variable-wise recombination probability p_v. In a binary-coded GA to handle real-parameter variables, two parent strings representing all n variables are usually operated by a single or a multi-point crossover procedure. If viewed from the point of the number of variables that get affected by such a recombination operation, it can be concluded that on an average $n/2$ variables are different when a parent solution is compared with an offspring solution. Among them, some are completely new and some come directly from the second parent. Argued in this manner, we can then suggest to use $p_v = \frac{n/2}{n} = 0.5$ for a real-parameter blending operation in which 50% variables would get recombined on an average. One way to implement this would be to recombine each variable with a probability $p_v = 0.5$ at a time. Another approach would be to choose 50% variables at random and then perform the chosen blending operation to each of the chosen variable one at a time. Due to their variable-by-variable operation, we call these operators as *variable-wise* recombination operators. Another approach for a real-parameter recombination would be to consider each parent as a point (or a vector) in the n-dimensional space and then to use a vector operation between two or more parent solutions to create a new offspring vector in the decision space. We call these operators as *vector-wise* recombination here.

In addition to a real-parameter recombination being a variable or a vector-wise operator, the existing recombination operators can also be classified based on their emphasis of creating offspring solutions near parents or near the centroid of the parents. We argue here that this aspect of recombination operators makes them *functionally* different from each other and may act as a critical distinguishing factor to make a significant effect on the performance of the overall algorithm. It is needless to say that both variable-wise and vector-wise recombination operators can be classified according to parent-centric or mean-centric operators. We now classify a number of existing real-parameter recombination operators into these two classes.

2.1 Parent-centric Recombination Operators

As the name suggests, in these operators, the offspring values or vectors are created around the parent values or vectors. The motivation behind these operators is that since the parent solutions are tested to be better in the preceding selection operation (as opposed to any other solution, such as the centroid of parent solutions, for example), creating offspring solutions near to parents may lead to a better solution. This is particularly true in the beginning of a run, when population members are initialized in the entire search space and are expected to be far away from each other. It then makes sense to create offspring solutions close to the parent solutions that are already considered to be better in the population by the selection operation. Other ideas, such as creating a solution close to the mean of the participating parent solutions, may be risky as such solutions may not have been tested for their objective and/or constraint values before offspring solutions are created near them. In the following, we describe two such real-parameter recombination operators, but it is worth mentioning here that a few other parent-centric recombination operator implementations exist in the literature.

2.1.1 Fuzzy Recombination (FR) Operator

Fuzzy-recombination is a variable-wise recombination operator and was proposed by Voigt et al. [16]. For two parent values (p_1 and p_2) of a particular variable, solutions close to each parent is chosen with a triangular probability distribution, as shown in Figure 1. The triangular probability

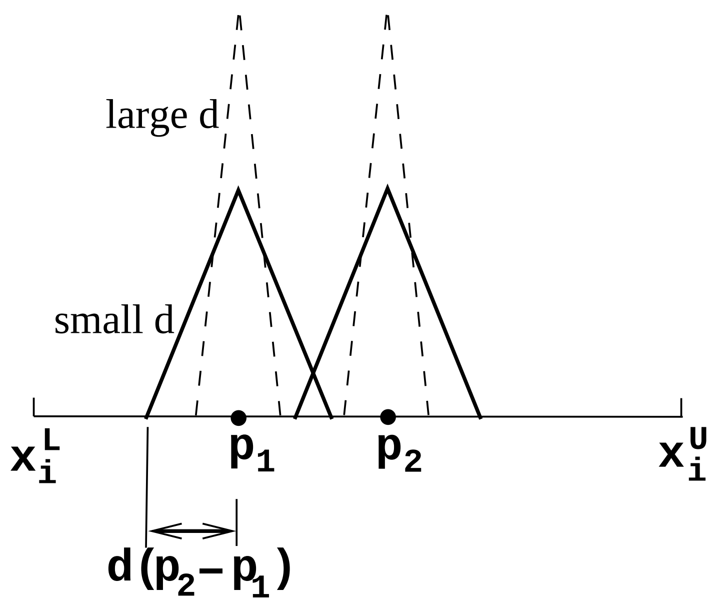


Figure 1: Fuzzy recombination (FR-d) operator, where d is a user-defined control parameter.

distribution emphasizes values close to each parent more likely to be created as offspring than values away from parents. This is the reason this operator qualifies to be a parent-centric recombination operator. The parameter d is introduced to further control the diversity of the created offspring values compared to the participating parent values. If d is chosen to be a small number, offsprings are created close to the parent values and vice versa. The effect of d on the probability distribution is shown in the figure.

However, even for a fixed d, if two parent values lie far away from each other, the difference between a created offspring value to its nearest parent value can be large, whereas if the two participating parent values are close to each other, the created offspring value is also likely to be close to one of the parent values. This dual control of offspring values vis-a-vis their participating parent values provides both algorithmic and user-specific control on the creation of the offspring population.

2.1.2 Simulated Binary Crossover (SBX) Operator

The simulated binary crossover (SBX) operator is also a variable-wise recombination operator that was proposed elsewhere [2]. This operator is similar to FR operator, but has the needed *ergodic* property [12] in order to have a non-zero probability of creating any number in the search space by recombining any two parent values from the same search space. For two representative parent values (p_1 and p_2), the SBX probability distribution is shown in Figure 2. This operator also possess

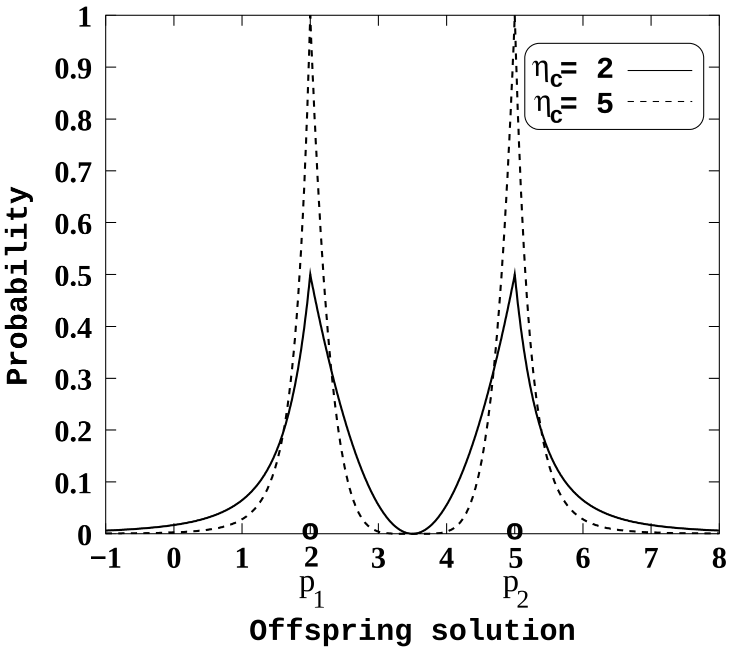


Figure 2: Simulated binary recombination (SBX-η) operator, where η is a user-defined control parameter.

the dual control on the diversity of created offspring values (c_1 and c_2) vis-a-vis the parent solutions that is mentioned above. The algorithmic control comes from the fact that the probability is defined on a non-dimensional variable, called the spread factor $\beta = |c1 - c2|/|p1 - p2|$. The user-specific control is achieved by means of a parameter η_c. The effect of η_c is shown in the figure.

In this study, we have modified this operator to achieve a self-adaptive parent to mean-centric property. For this reason, we describe this operator in somewhat more detail. First, the generic probability distribution to create an offspring value is defined as a function of a non-dimensionalized parameter $\beta \in [0, \infty]$ as follows:

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

Along with the mean-preserving property of offspring values and parent values ($c_1 + c_2 = p_1 + p_2$), the above probability distribution help preserve the following properties:

1. Both offspring values are equi-distant from parent values.

2. There exist a non-zero probability to create an offspring value anywhere in the entire real-space from any two parent values, except when both parent values are identical.

3. A recombination with $\beta = 1$ signifies that offspring and parents are identical.

4. A recombination with $\beta < 1$ indicates that offspring values are bracketed by the parent values.

5. A recombination with $\beta > 1$ indicates that offspring values are outside of the range of the parent values.

6. The overall probability of creating a pair of offspring values within the range of parent values is identical to overall probability of creating two offspring values outside the range of parent values.

Simple calculations will show that for the two participating parent values (p_1 and p_2), two offspring values (c_1 and c_2) can be created as a linear combination of parent values for an event with a random number $u \in [0, 1]$, as follows:

$$\begin{aligned} c_1 &= 0.5(1 + \beta(u))p_1 + 0.5(1 - \beta(u))p_2, & (2)\\ c_2 &= 0.5(1 - \beta(u))p_1 + 0.5(1 + \beta(u))p_2. & (3) \end{aligned}$$

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

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

The resulting weights to p_1 and p_2 are biased in such a way that offspring values close to the parent values are more likely than offspring values away from them, thereby providing SBX its parent-centric property.

The above description of the SBX operation can create offspring values anywhere in the real-space. For an optimization problem having no variable bound, this is a desired property. However, in most practical problems, every variable is usually bounded within a lower ($a = x_i^{(L)}$) and upper ($b = x_i^{(U)}$) bound. In a such a case, Equation 4 can be modified in a manner so as to have zero probability outside the specified range. For $p_1 < p_2$, lower bound a is closer to p_1 than to p_2, and we compute a parameter $\gamma_a = 1/(2\beta_a^{\eta_c+1})$, where $\beta_a = 1 + (p_1 - a)/(p_2 - p_1)$. We now calculate $\beta(u, a)$ as a function of the random number $u \in [0, 1]$ as follows:

$$\beta(u,a) = \begin{cases} (2u(1-\gamma_a))^{\frac{1}{\eta_c+1}}, & \text{if } u \le 0.5/(1-\gamma_a), \\ \left(\frac{1}{2(1-u(1-\gamma_a))}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise.} \end{cases} \tag{5}$$

The first offspring c_1 near p_1 is then calculated as follows:

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

Interestingly, when the bound a is at negative infinity (thereby indicating no finite lower bound), $\beta_a = \infty$ and $\gamma_a = 0$. In this case, $\beta(u, 0)$ computed from Equation 5 becomes exactly equal to $\beta(u)$ computed using Equation 4.

Similarly, γ_b may be computed in a similar manner except β_a must be replaced by $\beta_b = 1 + (b - p_2)/(p_2 - p_1)$ and $\beta(u, b)$ is computed by Equation 5 by replacing γ_a with γ_b. Then, the following equation can be used to compute the second offspring:

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

Figure 3 shows how the bounded-SBX operator does not allow values outside the range $[x_i^{(L)}, x_i^{(U)}]$ to be created and still prefers creating values close to the parent values.

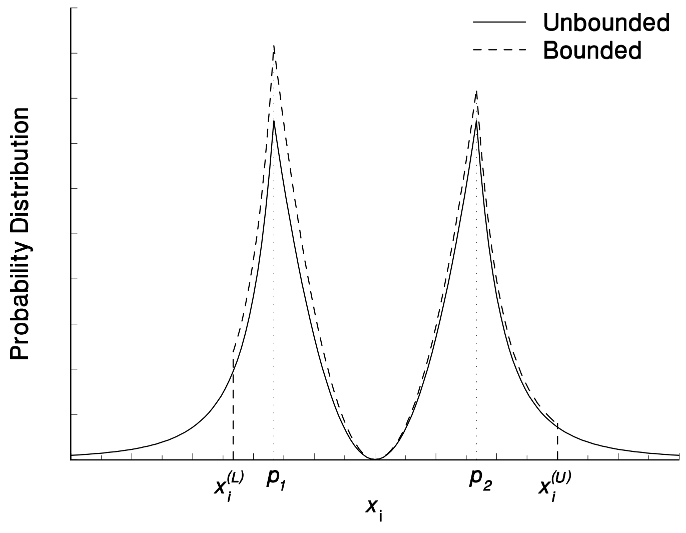


Figure 3: Bounded SBX operator (shown in dashed line) applied to two parent values p_1 and p_2 avoids creating an offspring outside the variable bounds $[x_i^{(L)}, x_i^{(U)}]$, but maintains a parent-centric approach.

2.1.3 PCX Operator

Parent-centric concept can also be extended to vector-wise recombination. One such operator is the parent-centric crossover or PCX suggested elsewhere [4]. In this operator, three parent solutions (vectors) are chosen from the GA population and a biased linear combination of them is proposed to create an offspring solution ($\vec{y}$) around one of the parent vectors ($\mathbf{x}^{(p)}$), as follows:

$$\vec{y} = \mathbf{x}^{(p)} + w_\zeta \|\mathbf{d}^{(p)}\| \vec{e}^{(p)} + \sum_{i=1,\ i\neq p}^{n} w_\eta \bar{D} \vec{e}^{(i)}, \tag{8}$$

where $\vec{e}^{(i)}$ are the n orthonormal bases that span the subspace. The direction $\vec{e}^{(p)}$ is along $\mathbf{d}^{(p)}$. The parameters w_ζ and w_η are zero-mean normally distributed variables with variance σ_ζ^2 and σ_η^2, respectively. The operator is described pictorially in Figure 4 by creating a number of offspring solutions from three specific parent solutions. The crowding of offspring solutions near one of the parent solution clearly indicate that PCX is a parent-centric recombination operator. The location of the other two parents control the distribution of the offspring solution, similar to that in SBX or in FR operator and the user-specified parameters w_ζ and w_η allows an external control of the diversity of offspring solutions.

2.2 Mean-Centric Crossover Operators

We now discuss the mean-centric recombination operators. In these operators, offspring are created around the centroid of the participating parents. The motivation for these operators come from the realization that since all parents were judged to be better by the selection operator, the intermediate region surrounded by parents may also be good. A few existing mean-centric recombination operators are described below.

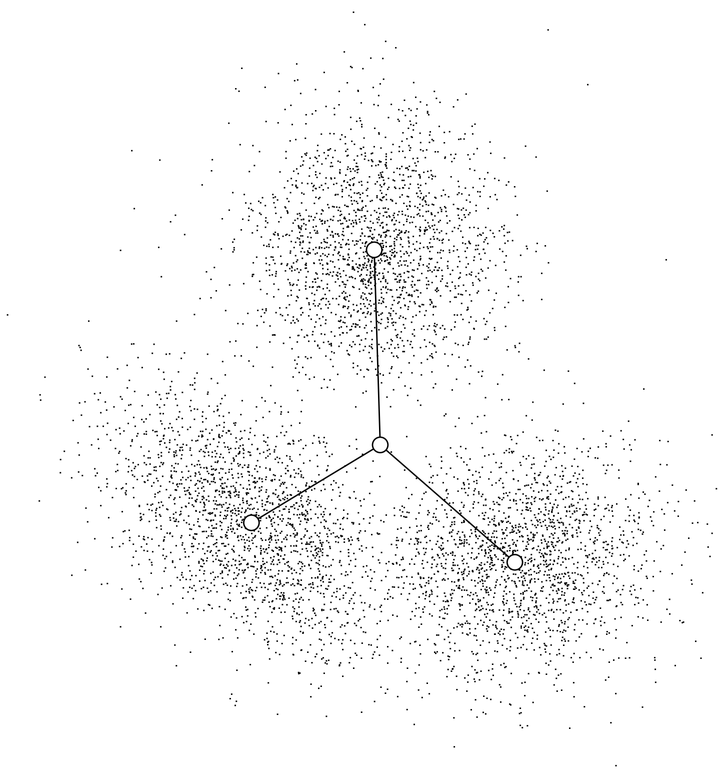


Figure 4: Parent-centric crossover (PCX) operator with three parent solutions.

2.2.1 Blend Crossover (BLX) Operator

BLX-α is a variable-wise recombination operator proposed by Eshelman and Schaffer [6]. Offspring values are created uniformly around the two parent values. A parameter α allows offspring to be created within or outside the range of parents. Figure 5 shows BLX operator for $\alpha = 0.5$ with a solid line, in which values in the range 50% outside the region described by the parents, are chosen at random. For a large α, the created offspring value may be far from the parent values.

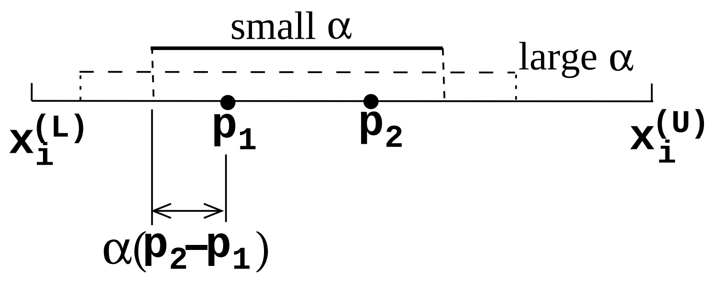


Figure 5: BLX-α operator, where α is a user-defined control parameter.

2.2.2 Unimodal Normally Distributed Crossover (UNDX) Operator

In the unimodal normally distributed crossover (UNDX) operator [11] (Figure 6), three or more parents are chosen from the GA population and points around the centroid of the parents are created with a normal distribution. A couple of inherent parameters control the distribution of the offspring solutions, similar to that in the PCX operator. Thus, UNDX operator is a vector-wise mean-centric recombination operator.

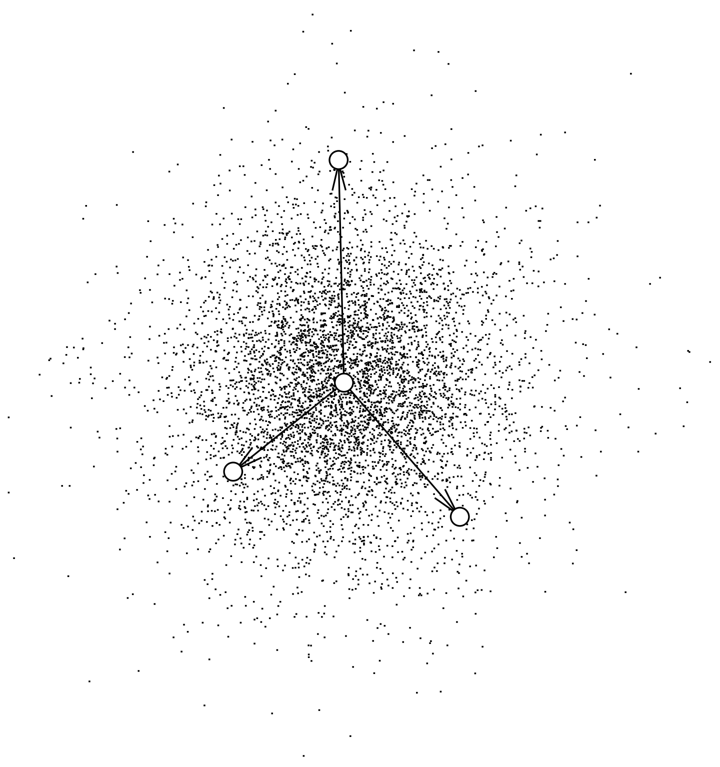


Figure 6: Unimodal normally distributed crossover (UNDX) operator with three parent solutions.

3 Parent or Mean-Centric Recombination

The above subsections demonstrate that both types of recombination operators – parent-centric and mean-centric – can exist in both variable-wise and vector-wise forms, but it is important to understand under what scenarios each type of recombination operator will be more effective. Let us investigate a couple of scenarios for this purpose.

Figure 7 shows Scenario 1 in which an EA population is approaching the optimum. This scenario is likely to happen early on in a simulation in which the current population as a whole does not bracket the optimum solution and GA operators must decide how the new population members must be created so as to progress towards the optimum solution. In such a scenario, the population-best solution is likely to lie at the boundary of the current population. Although this need not be the case, particularly in handling multi-modal problems in which the current population-best solution, attracted by a local optimal solution, may lie well inside the population, while the whole population can be away from the global optimal solution. Although this is a possibility, we do not base our algorithm on such an exception here and hope that GA's operators, although getting attracted to a local optimal solution, can find a new and better solution out of the basin of attraction of the current local optimal solution. Ideally, in such a scenario (when the population-best solution is at the boundary of the current population), a parent-centric recombination is likely to yield better offspring solutions, as solutions beyond the population-best solution and are likely to be closer to the optimum. Thus, while approaching an optimum, it may be a better strategy to employ a

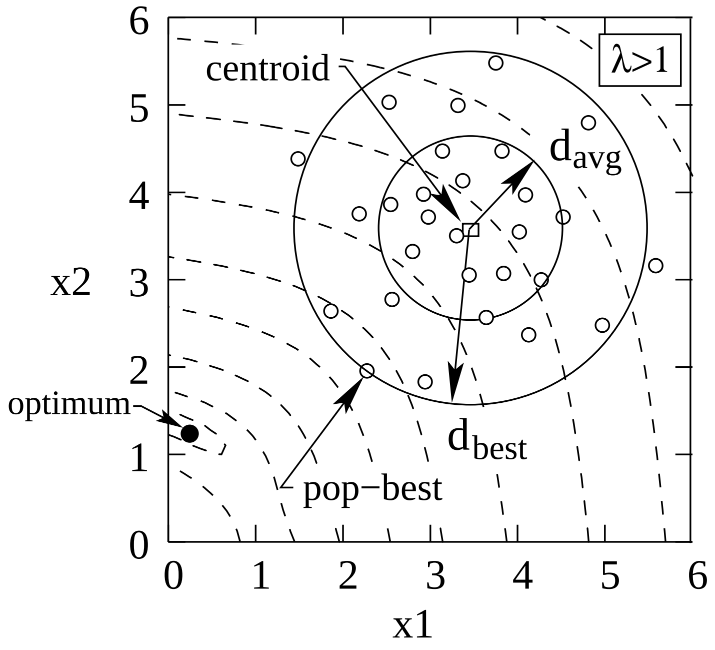


Figure 7: Scenario 1: Population is approaching the optimum.

parent-centric recombination operator.

On the other hand, Figure 8 shows the Scenario 2 in which the GA population surrounds the optimum and the population-best solution is likely to be an intermediate population member. In such a case, a mean-centric recombination operator is likely to produce better solutions.

While the above two scenarios and corresponding nature of recombination operations are somewhat clear, it is not clear how to know which scenario exist at a particular generation so that a suitable type of recombination operator can be applied. In this study, we attempt to identify one of the two scenarios by computing a parameter λ for this purpose:

$$\lambda = \frac{d_{\text{best}}}{d_{\text{avg}}}, \tag{9}$$

where d_{best} is Euclidean distance of population-best solution from the centroid of the population in the variable space and d_{avg} is the average distance of population members from the centroid of the population. A little thought and an inspection of the above two figures will reveal that in the first scenario, λ is likely to be greater than one, since the population-best solution is likely to lie at the edge of the population. However, in the second scenario, it is likely that the parameter λ will be smaller than one. Since the parameter λ hopes to capture the status of a population's approach towards the optimum or convergence near the optimum, we use this parameter to redefine the creation of offspring in a self-adaptive way in the SBX operator. We discuss this redefinition in the following section.

4 Proposed Self-Adaptive SBX Operator

To modify the SBX operator, first, we calculate the parameter λ for the current population. Now, for each recombination operation involving two parent values p_1 an p_2, we redefine two virtual

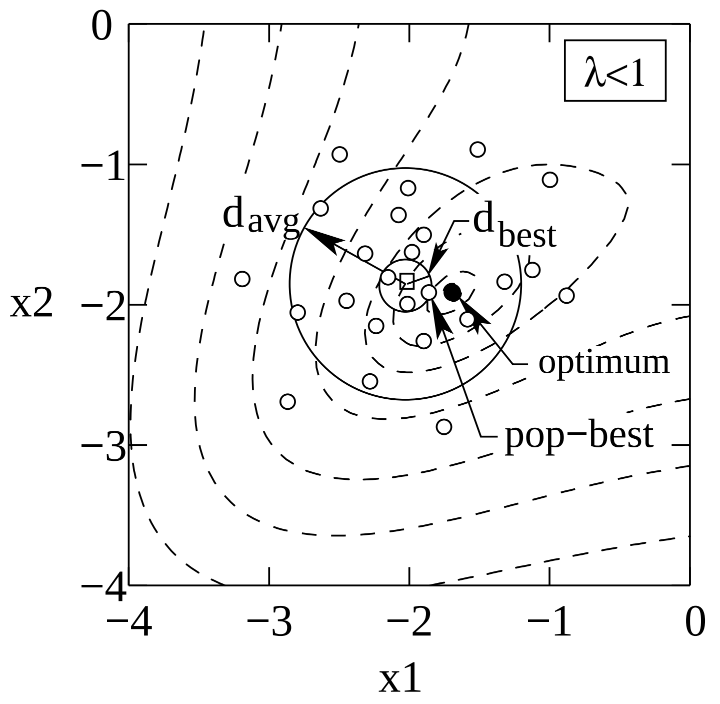


Figure 8: Scenario 2: Population surrounds the optimum.

parents v_1 and v_2, as follows:

$$v_1 = \frac{p_1 + p_2}{2} - \lambda\frac{p_2 - p_1}{2}, \quad (10)$$

$$v_2 = \frac{p_1 + p_2}{2} + \lambda\frac{p_2 - p_1}{2}. \quad (11)$$

Next, the virtual parents (instead of p_1 and p_2) are then used to create two offspring values performing a SBX operation using equations 2 and 3. The above modification to the SBX operation is simple, but let us now understand how the λ parameter can make SBX's inherent parent-centric property to act more like a mean-centric property for certain λ values.

First, let us consider $\lambda = 1$. This will make $v_1 = p_1$ and $v_2 = p_2$, meaning that virtual parents are identical to the actual parents and that the original SBX operation takes place. Let us now consider both the scenarios discussed in Figures 7 and 8. In Scenario 1, $\lambda > 1$ (population-best variable value lies outside the range of two parent values) and above equations suggest that two virtual parents are also created outside the range of the parents. Figure 9 shows the location of virtual parents for a given pair of parents. The dashed line shows the original SBX probability distribution and the solid line shows the modified SBX probability distribution. Clearly, more emphasis is now given to regions outside the region bounded by the original parents. In such scenarios, more points will be created outside the region of current population and the population has a better chance of progressing towards the optimum quicker.

For Scenario 2, the opposite happens. The parameter $\lambda < 1$ and the virtual parents will lie inside the region bounded by the parents, as shown in Figure 10. The solid line shows that now there is a large probability of creating offspring solutions inside the parents. This property of the modified SBX operator will provide it a mean-centric property which was not possible to be achieved with the original SBX operator. It is clear from these discussions that by changing the λ parameter the modified SBX operator can introduce parent or mean-centric property to the operator. A small

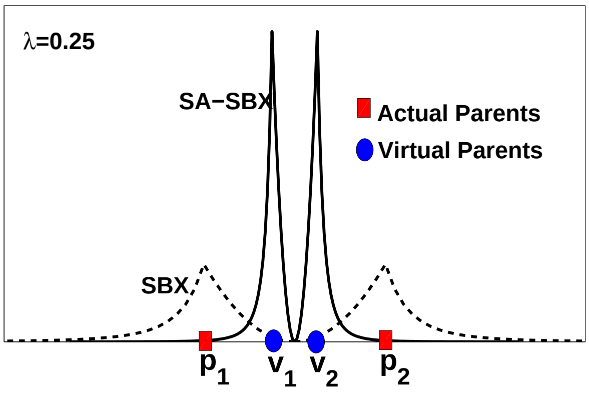


Figure 9: Virtual parents lie outside the range of parents in Scenario 1.

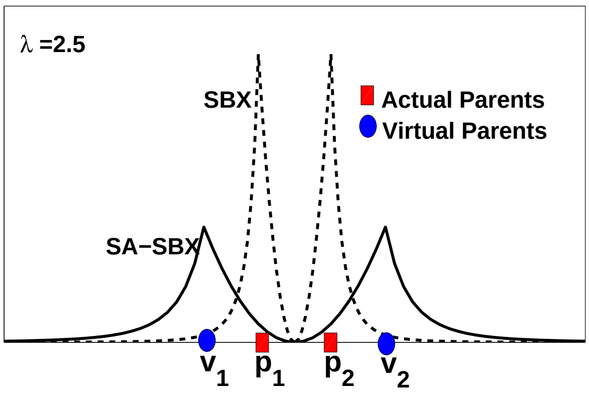


Figure 10: Virtual parents lie inside the range of parents in Scenario 2.

value of λ will cause the operator to behave like a mean-centric operator and a large value will cause it to act like a parent-centric operator and $\lambda \approx 1$ keeps it similar to the original SBX operator.

To quantify the extent of mean-centric operation introduced by a λ value, we compute the overall probability of creating an offspring value within the parent values as a function of λ. Figure 11 shows that for $\lambda = 0$, all offspring solutions are created inside the parent values. With an increase of λ, the probability reduces and at $\lambda = 1$, it is equal to 0.5, meaning that 50% offspring values lie inside the parent values for such a case. For λ beyond one, more offspring values are created outside the range of parents. This figure shows the controllability on the extent of mean-centric (for that matter, extent of parent-centric) property by the λ parameter. Thus, if the location of the current population with respect to the optimal solution is represented by an appropriate λ parameter, our modified SBX operator is ready to create an appropriate offspring solution to support the cause.

5 Results with SA-SBX Operator

We now present an extensive set of simulation results demonstrating the working of the proposed SA-SBX operator. To evaluate the effect of recombination operator alone, we have not used any mutation operator here. However, we have used the binary tournament selection operator to choose parents for the SA-SBX operator. No elite preservation technique is used and the algorithm is a generational GA in which the parent population is completely replaced by the offspring population of the same size.

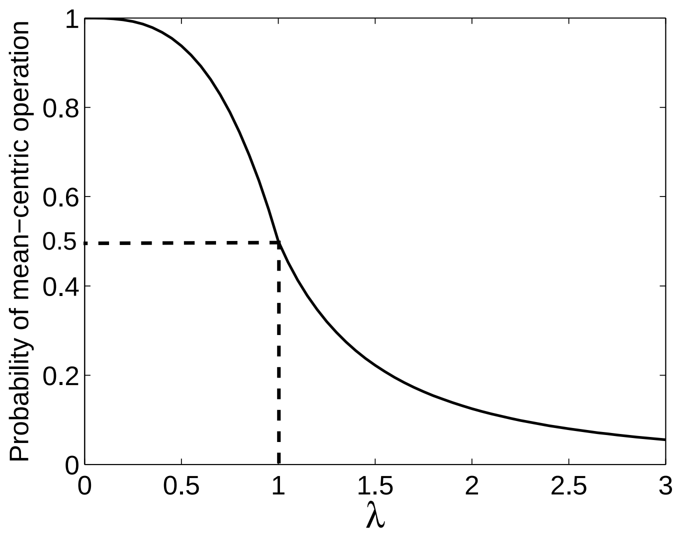


Figure 11: Probability of creating offspring values inside the parent values (the extent of mean-centric operation) as a function of λ.

We consider two sets of test problems. In the first set, we choose four unimodal and two multi-modal test problems from the EA literature. They are presented in Table 1. In the second set, we choose three unimodal and three multi-modal shifted and rotated test problems taken from the EA literature. They are presented in Table 2.

Table 1: Set 1: Unimodal and multi-modal test problems used in the study. All problems have an optimal objective value of $f^* = 0$.

Prob.	Objective function
	Unimodal Problems
P1	$f_{\text{ellipsoidal}} = \sum_{i=1}^{n} i x_i^2$
P2	$f_{\text{discus}} = 10^4 x_1^2 + \sum_{i=2}^{n} x_i^2$
P3	$f_{\text{cigar}} = x_1^2 + 10^4 \sum_{i=2}^{n} x_i^2$
P4	$f_{\text{ridge}} = x_1^2 + 100\sqrt{\sum_{i=2}^{n} x_i^2}$
	Multi-modal Problems
P5	$f_{\text{griewangk}} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}} + 1\right)$
P6	$f_{\text{schaffer}} = \left(\frac{1}{n-1}\sum_{i=1}^{n-1}\sqrt{s_i} + \sqrt{s_i}\sin^2(50 s_i^{1/5})\right)^2, \quad \text{where} \quad s_i = \sqrt{x_i^2 + x_{i+1}^2}$
P7	$f_{\text{rastrigin}} = \sum_{i=1}^{n}\left(x_i^2 + 10(1 - \cos(2\pi x_i))\right)$
P8	$f_{\text{ackley}} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + \exp(1)$

To compare the performance of the proposed self-adaptive SBX operator, we use two variations of the recombination operator: RGA with the original SBX operator (a purely parent-centric operator) and RGA with a purely mean-centric operator, in which two offspring points (c_1 and c_2) are created from two parents (p_1 and p_2) using the following probability distribution: $P(\beta) = (\eta + 1)/(\beta + 1)^{\eta+2}$, where $\beta = |(c_2 - c_1)/(p_2 - p_1)|$. In each case, 20 runs are made from different initial populations until a specified number of function evaluations are elapsed. Thereafter, best, median and worst objective value attained in 20 runs are tabulated and plotted. In all cases, a

Table 2: Set 2: Shifted and rotated unimodal and multi-modal test problems used in the study. Problems P9 to P12 are initialized as $x_i \in [-100, 100]$, P13 as $x_i \in [0, 600]$ and P14 as $x_i \in [\pi, \pi]$.

Prob.	Objective function
	Unimodal Problems
P9	$f_{\text{ssphere}} = \sum_{i=1}^{n} z_i^2 - 450, \quad \mathbf{z} = \mathbf{x} - \mathbf{o}$
P10	$f_{\text{sschwefel}} = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} z_j^2\right) - 450, \quad \mathbf{z} = \mathbf{x} - \mathbf{o}$
P11	$f_{\text{srellipsoidal}} = \sum_{i=1}^{n} i z_i^2 - 450, \quad \mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M}$
	Multi-modal Problems
P12	$f_{\text{srosenbrock}} = \sum_{i=1}^{n-1} \left(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2\right) + 390, \quad$ **z=(x-o)+1**
P13	$f_{\text{sgriewangk}} = \sum_{i=1}^{n} \left(\frac{z_i^2}{4000}\right) - \prod_{i=1}^{n} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 - 180, \quad$ **z=(x-o)*M**
P14	$f_{\text{multi-schwefel}} = \sum_{i=1}^{n} (\mathbf{A}_i - \mathbf{B}_i(\mathbf{x}))^2 - 460$ $\mathbf{A}_i = \sum_j (a_{ij} \sin(\alpha_j) + b_{ij} \cos(\alpha_j))$ $\mathbf{B}_i(\mathbf{x}) = \sum_j (a_{ij} \sin(x_j) + b_{ij} \cos(x_j))$ **A, B** are two $n \times n$ matrices, a_{ij}, b_{ij} are integer random numbers in $[-100, 100]^n$ and α_j are random numbers in $[-\pi, \pi]^n$

population of size $5n$ (where n is the number of variables in the problem), a crossover probability of $p_c = 0.9$, and $\eta_c = 2$ are followed.

To investigate the effect of the location of initial population on the performance of the respective algorithms, we have chosen two initialization processes: (i) Symmetric initialization around the optimum ($\mathbf{x}^*$): $x_i \in x_i^* + [-20, 20]$ for all variables, and (ii) One-sided initialization away from optimum: $x_i \in x_i^* + [20, 60]$ for all variables. Each problem is considered for different sizes, $n = 20$, 50, and 100. For the second set of test problems, we consider $n = 10$, 30 and 50, except for P14 for which only $n = 10$ is considered. None of the methods considered here could find a function value smaller than 1.0 on this problem with larger variable sizes.

5.1 Results on Set 1 Test Problems

First, we present the results on Set 1 test problems by three algorithms initialized with a symmetric initialization process discussed above. Tables 3 and 4 tabulate the median, best and worst values of the population-best f at the end of certain overall function evaluations (presented in column 3 of the table) for unimodal and multi-modal problems, respectively.

A run is terminated after these limiting function evaluations are completed. The population-best objective function value is indicated in columns 4, 5 and 6 for different algorithms. In each case, three different problem sizes are considered. A comparison of objective values presented in columns 4, 5 and 6 will reveal the fact that the proposed SA-SBX outperforms the RGA with parent-centric or mean-centric recombination operators alone. In these problems, the parent-centric RGA performed better than the mean-centric RGA, but on all problems (P1 to P8) the SA-SBX is better. Another observation is that the number of function evaluations needed to achieve a similar performance is more for larger-dimensional problems.

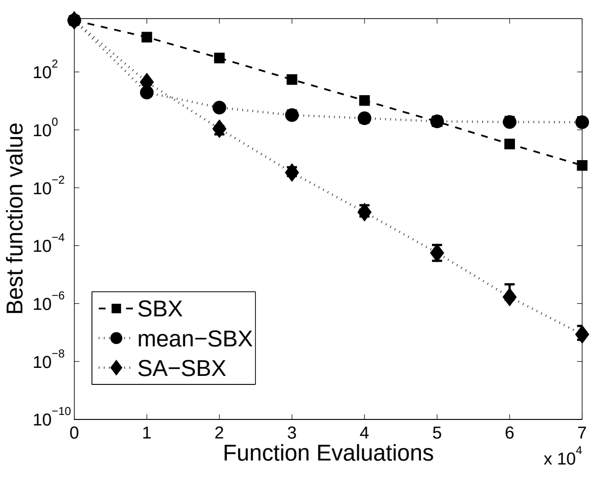
Figures 12(a), (b), (c) and (d) show the variation of the population-best objective value with generation counter for three representative problems – two unimodal (P2 and P4) and two multi-modal (P5 and P6) problems. In each case, the rate of convergence of SA-SBX is much better than the other two algorithms. Similar performances are observed for other problems, but they are not shown here for brevity. Figures 13(a), (b), (c) and (d) show similar results for the 100-variable version of the same three problems. The scalability of the proposed SA-SBX algorithm from 20 to 100-variable problems is clear from these results.

Table 3: Median, best and worst population-best objective function values (in first, second and third rows, respectively) obtained by three algorithms are shown for Set 1 unimodal problems with symmetric initialization ($x_i \in [-20, 20]$).

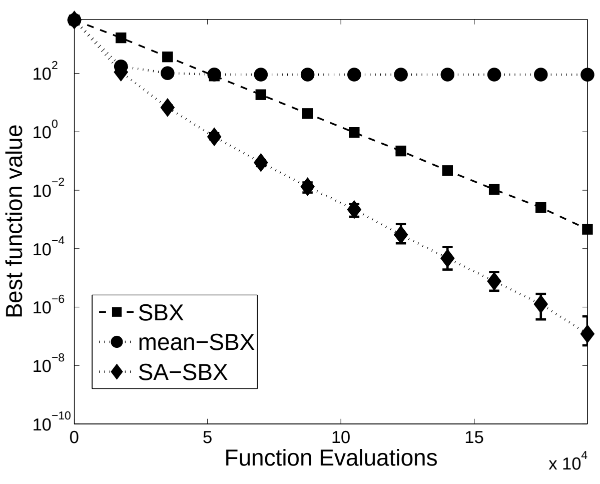
Prob.	n	FE	Parent-centric SBX	Mean-centric SBX	SA-SBX	M-SA-SBX
$P1$	20	20,500	5.59×10^{-6}	2.91	1.90×10^{-8}	$\mathbf{1.60 \times 10^{-8}}$
			1.79×10^{-6})	4.63×10^{-1}	1.00×10^{-8}	1.00×10^{-8}
			3.04×10^{-5}	9.62	1.79×10^{-6}	1.03×10^{-7}
	50	87,500	4.61×10^{-2}	2.13×10^{1}	$\mathbf{1.20 \times 10^{-8}}$	7.40×10^{-7}
			2.15×10^{-2}	1.03×10^{1}	1.00×10^{-8}	4.70×10^{-7}
			8.78×10^{-2}	3.63×10^{1}	4.80×10^{-8}	1.82×10^{-6}
	100	290,000	5.67×10^{3}	1.49×10^{2}	$\mathbf{1.10 \times 10^{-8}}$	1.26×10^{-5}
			3.30×10^{3}	9.57×10^{1}	1.00×10^{-8}	8.31×10^{-6}
			7.74×10^{3}	2.07×10^{2}	3.50×10^{-8}	2.07×10^{-5}
$P2$	20	19,000	7.78×10^{-6}	1.03	3.00×10^{-8}	$\mathbf{1.10 \times 10^{-8}}$
			2.91×10^{-6}	4.05×10^{-1}	1.00×10^{-8}	1.00×10^{-8}
			2.69×10^{-5}	6.26	3.32×10^{-6}	1.47×10^{-7}
	50	76,250	2.09×10^{-2}	1.82	$\mathbf{1.40 \times 10^{-8}}$	1.37×10^{-6}
			1.31×10^{-2}	9.28×10^{-1}	1.00×10^{-8}	5.87×10^{-7}
			4.29×10^{-2}	4.71	7.10×10^{-8}	3.53×10^{-6}
	100	245,000	3.70×10^{2}	5.77	$\mathbf{1.00 \times 10^{-8}}$	1.89×10^{-5}
			2.63×10^{2}	3.98	1.00×10^{-8}	1.42×10^{-5}
			4.62×10^{2}	9.34	2.00×10^{-8}	2.34×10^{-5}
$P3$	20	25,500	2.74×10^{-5}	2.18×10^{3}	1.03×10^{-6}	$\mathbf{1.50 \times 10^{-8}}$
			5.81×10^{-6}	4.69×10^{2}	1.00×10^{-8}	1.00×10^{-8}
			1.17×10^{-4}	5.79×10^{3}	6.09×10^{-4}	1.59×10^{-7}
	50	122,500	5.24×10^{-2}	1.01×10^{4}	9.53×10^{-7}	$\mathbf{1.90 \times 10^{-8}}$
			2.63×10^{-2}	5.82×10^{3}	8.40×10^{-8}	1.00×10^{-8}
			1.19×10^{-1}	1.57×10^{4}	3.46×10^{-5}	3.90×10^{-8}
	100	440,000	1.15×10^{5}	3.55×10^{4}	1.25×10^{-6}	$\mathbf{1.20 \times 10^{-8}}$
			8.36×10^{4}	2.18×10^{4}	1.67×10^{-7}	1.00×10^{-8}
			1.85×10^{5}	4.69×10^{4}	3.28×10^{-5}	2.60×10^{-8}
$P4$	20	38,000	5.27×10^{-6}	4.35×10^{1}	1.07×10^{-6}	$\mathbf{1.80 \times 10^{-8}}$
			3.09×10^{-6}	1.66×10^{1}	5.90×10^{-8}	1.00×10^{-8}
			1.25×10^{-5}	1.18×10^{2}	2.13×10^{-4}	5.60×10^{-8}
	50	185,000	9.01×10^{-4}	9.08×10^{1}	3.25×10^{-7}	$\mathbf{1.40 \times 10^{-8}}$
			6.22×10^{-4}	5.77×10^{1}	4.90×10^{-8}	1.00×10^{-8}
			1.50×10^{-3}	1.34×10^{2}	7.29×10^{-5}	2.80×10^{-8}
	100	657,500	6.46×10^{1}	1.76×10^{2}	3.38×10^{-7}	$\mathbf{1.40 \times 10^{-8}}$
			5.56×10^{1}	1.42×10^{2}	7.90×10^{-8}	1.00×10^{-8}
			8.43×10^{1}	2.13×10^{2}	1.26×10^{-6}	2.40×10^{-8}

Table 4: Median, best and worst population-best objective function values (in first, second and third rows, respectively) obtained by three algorithms are shown for Set 1 multi-modal problems with symmetric initialization ($x_i \in [-20, 20]$).

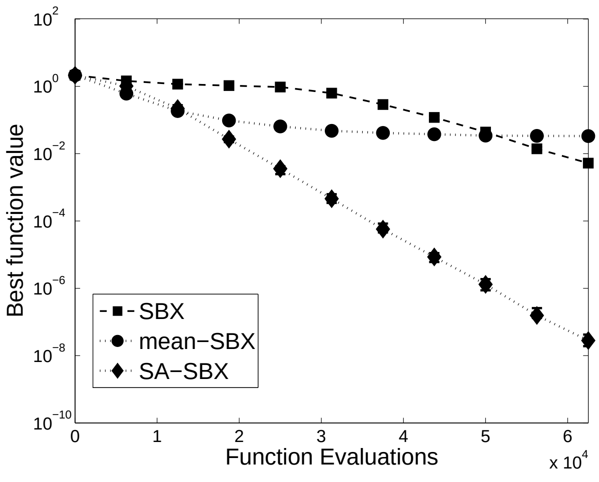
Prob.	n	FE	Parent-centric SBX	Mean-centric SBX	SA-SBX	M-SA-SBX
$P5$	20	17,000	8.42×10^{-7}	2.29	$\mathbf{4.00 \times 10^{-8}}$	9.70×10^{-8}
			1.46×10^{-7}	1.65	1.00×10^{-8}	1.00×10^{-8}
			7.40×10^{-3}	2.78	1.97×10^{-2}	3.43×10^{-2}
	50	65,000	3.30×10^{-3}	3.31×10^{-2}	$\mathbf{1.20 \times 10^{-8}}$	9.38×10^{-7}
			2.00×10^{-3}	1.46×10^{-2}	1.00×10^{-8}	4.06×10^{-7}
			5.50×10^{-3}	4.84×10^{-2}	3.00×10^{-8}	9.90×10^{-3}
	100	200,000	1.14	5.20×10^{-2}	$\mathbf{1.00 \times 10^{-8}}$	9.75×10^{-6}
			1.10	2.93×10^{-2}	1.00×10^{-8}	5.71×10^{-6}
			1.22	7.88×10^{-2}	1.40×10^{-8}	1.42×10^{-5}
$P6$	20	23,000	1.34×10^{-6}	3.69×10^{-2}	$\mathbf{1.00 \times 10^{-8}}$	7.00×10^{-8}
			7.47×10^{-7}	1.90×10^{-2}	1.00×10^{-8}	2.50×10^{-8}
			2.62×10^{-6}	1.05×10^{-1}	3.70×10^{-8}	1.35×10^{-7}
	50	90,000	1.26×10^{-4}	7.59×10^{-2}	$\mathbf{1.20 \times 10^{-8}}$	1.85×10^{-6}
			5.11×10^{-5}	1.55×10^{1}	1.00×10^{-8}	1.01×10^{-6}
			9.52×10^{-5}	2.25×10^{1}	3.90×10^{-8}	2.56×10^{-6}
	100	270,000	2.46×10^{-2}	1.15×10^{-1}	$\mathbf{1.10 \times 10^{-8}}$	1.95×10^{-5}
			2.12×10^{-2}	8.89×10^{-2}	1.00×10^{-8}	1.18×10^{-5}
			2.96×10^{-2}	1.43×10^{-1}	2.10×10^{-8}	2.51×10^{-5}
$P7$	20	60,000	7.95	2.05×10^{1}	7.96	**5.97**
			2.98	1.26×10^{1}	4.97	2.98
			1.29×10^{1}	3.62×10^{1}	1.69×10^{1}	8.95
	50	180,000	1.18×10^{2}	4.63×10^{1}	**2.98**	**2.98**
			1.82×10^{1}	3.65×10^{1}	1.99	1.99
			2.48×10^{2}	6.00×10^{1}	6.96	6.96
	100	450,000	1.21×10^{3}	1.21×10^{2}	2.38×10^{1}	**4.97**
			1.08×10^{3}	9.81×10^{1}	1.29×10^{1}	9.95×10^{-1}
			1.29×10^{3}	1.53×10^{2}	2.79×10^{1}	9.95
$P8$	20	30,000	3.92×10^{-6}	1.39	8.10×10^{-8}	$\mathbf{4.90 \times 10^{-8}}$
			1.65×10^{-6}	2.53×10^{-1}	1.00×10^{-8}	1.50×10^{-8}
			6.67×10^{-6}	2.65	2.30×10^{-3}	1.61×10^{-7}
	50	120,000	1.00×10^{-3}	1.46	$\mathbf{3.80 \times 10^{-8}}$	9.09×10^{-7}
			6.70×10^{-4}	7.26×10^{-1}	1.50×10^{-8}	5.33×10^{-7}
			1.60×10^{-3}	2.18	1.07×10^{-7}	1.49×10^{-6}
	100	390,000	4.11	1.79	$\mathbf{1.50 \times 10^{-8}}$	2.99×10^{-6}
			3.61	1.52	1.00×10^{-8}	2.10×10^{-6}
			4.62	2.07	2.20×10^{-8}	4.32×10^{-6}



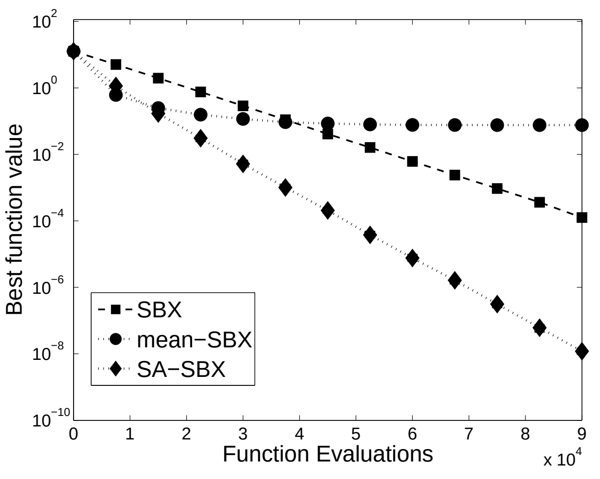
(a) P2



(b) P4



(c) P5

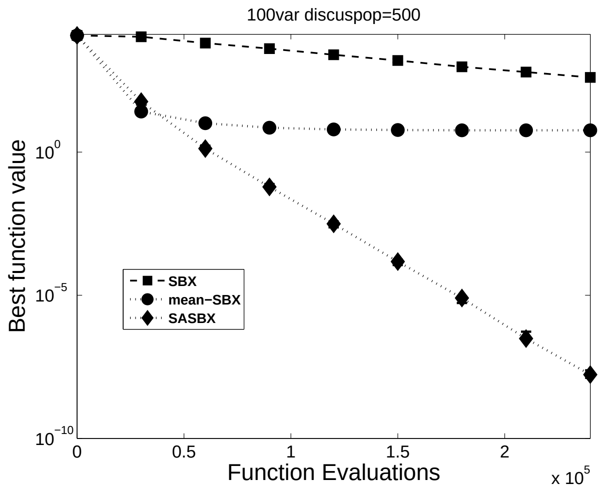


(d) P6

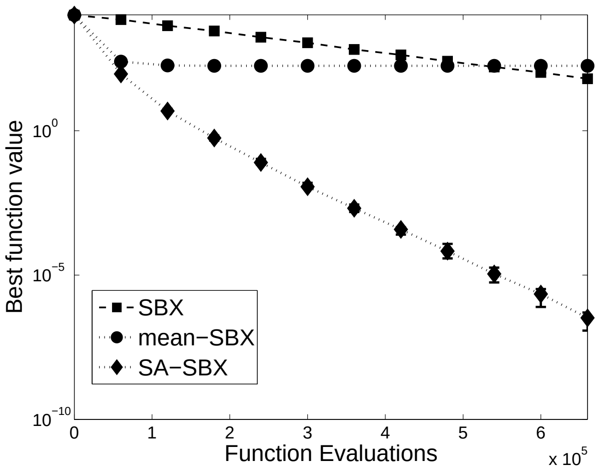
Figure 12: Population-best f for 50-variable problems P2, P4, P5 and P6 obtained using RGAs with SA-SBX, original parent-centric SBX, and mean-centric operator initialized symmetrically.

To investigate the working of the proposed adaptive recombination approach, we plot the variation of the parameter λ with generation for a typical run on problem P1 in Figure 14. Since the initial population already surrounds the optimum in a symmetric initialization, λ is expected to be less than one. Importantly, the RGA with SA-SBX reduces λ with generations, as evident from the figure. It is interesting that original SBX operator would have kept $\lambda = 1$ in all generations, but now the proposed SA-SBX operator uses a varying λ to help converge to the optimum in a computationally fast manner. After sufficient number of generations when population is close to optimum there are large fluctuations in λ value, this is attributed to that fact that near to optimum the entire population shrinks to a very small region hence both d^c_{avg} and d^c_{best} become very small to give fluctuating values of λ.

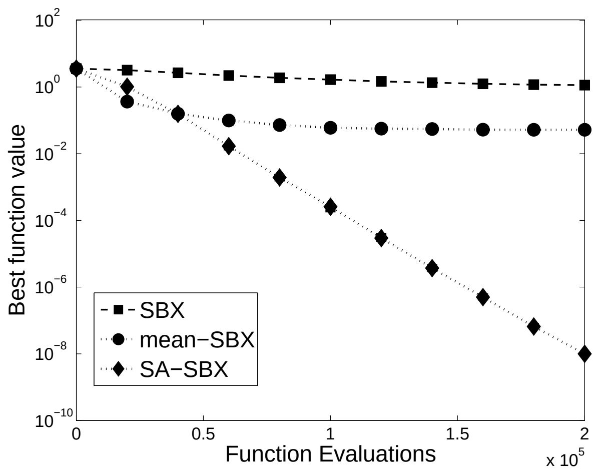
The middle sub-figure shows that the difference between the average population-distance from centroid and population-best point from centroid is positive, thereby indicating that population-best solution is closer to the centroid of the population throughout the simulation run than a random population member. Eventually with generations, the difference comes close to zero, indicating the converging property of the population with generation. The bottom-most sub-figure shows the difference between distance of the centroid from the known optimum and the distance of the population-best solution from optimum. The negative values indicate that the population-best solution is somewhat away from the true optimum than the population-centroid in initial genera-



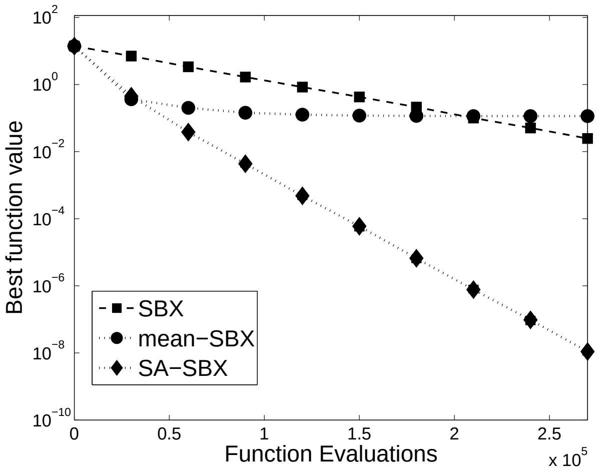
(a) P2



(b) P4



(c) P5



(d) P6

Figure 13: Population-best f for 100-variable problems P2, P4, P5 and P6 obtained using RGAs with SA-SBX, original parent-centric SBX, and mean-centric operator initialized symmetrically.

tions, but eventually both distances become equal, thereby demonstrating the converging property of the algorithm.

Tables 5 and 6 tabulate the final population-best objective function values for Set 1 test problems solved using the same three algorithms but this time initialized using one-sided initialization process ($x_i \in [20, 60]$).

Due to the creation of the initial population away from the optimum, we have allowed somewhat more function evaluations before terminating a run. The table shows a similar superior performance of SA-SBX operator compare to other two operators used in the study.

Figures 15 and 16 show the convergence pattern of the three algorithms for problems P2, P4, P5 and P6. The variation of population-best objective value over 25 different runs shows reliable performance of the proposed procedure.

Since P1 is a unimodal problem, in early generations, one-sided initialization makes one of the boundary points as the population-best point. Hence, λ is expected to be greater than one early on. Figure 17 verifies this fact. Interestingly, this trend is opposite to that observed in the symmetric initialization case (Figure 14). An increased λ enables a larger diversity in offspring solutions thereby causing a faster approach towards the optimum. When the population comes around the optimum, a situation similar to the symmetric initialization happens and the algorithm

Table 5: Median, best and worst population-best objective function values (in first, second and third rows, respectively) obtained by three algorithms are shown for Set 1 unimodal problems with one-sided initialization ($x_i \in [20, 60]$).

Prob.	n	FE	Parent-centric SBX	Mean-centric SBX	SA-SBX	M-SA-SBX
$P1$	20	23,000	2.51×10^{-5}	4.45×10^{4}	1.89×10^{-6}	$\mathbf{2.40 \times 10^{-8}}$
			8.15×10^{-6}	2.68×10^{4}	2.30×10^{-8}	1.00×10^{-8}
			7.91×10^{-5}	5.44×10^{4}	6.36×10^{-4}	3.02×10^{-7}
	50	111,250	1.04×10^{-2}	3.39×10^{5}	$\mathbf{1.30 \times 10^{-8}}$	5.90×10^{-8}
			5.80×10^{-3}	2.29×10^{5}	1.00×10^{-8}	2.50×10^{-8}
			1.72×10^{-2}	3.98×10^{5}	1.11×10^{-7}	2.66×10^{-7}
	100	370,000	8.17×10^{3}	1.77×10^{6}	$\mathbf{1.30 \times 10^{-8}}$	8.33×10^{-7}
			4.50×10^{3}	1.54×10^{6}	1.00×10^{-8}	5.25×10^{-7}
			1.54×10^{4}	1.98×10^{6}	4.20×10^{-8}	1.93×10^{-6}
$P2$	20	22,000	2.84×10^{-5}	1.59×10^{5}	2.12×10^{-6}	$\mathbf{2.40 \times 10^{-8}}$
			6.67×10^{-6}	1.24×10^{4}	5.80×10^{-8}	1.00×10^{-8}
			2.75×10^{-4}	3.81×10^{6}	1.42×10^{5}	3.91×10^{-7}
	50	102,500	4.20×10^{-3}	3.30×10^{4}	$\mathbf{1.20 \times 10^{-8}}$	3.70×10^{-8}
			3.10×10^{-3}	2.32×10^{4}	1.00×10^{-8}	1.50×10^{-8}
			1.11×10^{-2}	6.13×10^{4}	6.60×10^{-8}	1.04×10^{-7}
	100	327,500	5.73×10^{2}	6.23×10^{4}	$\mathbf{1.00 \times 10^{-8}}$	7.70×10^{-7}
			3.93×10^{2}	5.18×10^{4}	1.00×10^{-8}	4.91×10^{-7}
			7.66×10^{2}	9.21×10^{4}	3.40×10^{-8}	1.40×10^{-6}
$P3$	20	28,500	6.30×10^{-5}	3.95×10^{7}	2.04×10^{-4}	2.70×10^{-8}
			1.15×10^{-5}	2.56×10^{7}	9.95×10^{-7}	1.00×10^{-8}
			2.75×10^{-4}	6.81×10^{7}	6.19	8.80×10^{-8}
	50	137,500	5.63×10^{-2}	1.44×10^{8}	4.54×10^{-5}	$\mathbf{1.30 \times 10^{-8}}$
			2.50×10^{-2}	1.23×10^{8}	8.06×10^{-7}	1.00×10^{-8}
			1.03×10^{-1}	1.75×10^{8}	3.92×10^{-4}	5.00×10^{-8}
	100	485,000	3.52×10^{5}	4.08×10^{8}	3.63×10^{-5}	$\mathbf{1.30 \times 10^{-8}}$
			1.91×10^{5}	3.80×10^{8}	3.99×10^{-6}	1.00×10^{-8}
			5.88×10^{5}	4.42×10^{8}	4.64×10^{-4}	2.80×10^{-8}
$P4$	20	41,500	6.56×10^{-6}	7.18×10^{3}	5.09×10^{-6}	$\mathbf{1.60 \times 10^{-8}}$
			1.82×10^{-6}	6.42×10^{3}	2.28×10^{-7}	1.00×10^{-8}
			1.75×10^{-5}	8.64×10^{3}	4.20×10^{-3}	8.30×10^{-8}
	50	198,750	1.10×10^{-3}	1.27×10^{-4}	1.15×10^{-6}	$\mathbf{1.50 \times 10^{-8}}$
			7.50×10^{-4}	1.18×10^{4}	4.45×10^{-7}	1.00×10^{-8}
			1.60×10^{-3}	1.46×10^{4}	7.75×10^{-5}	2.30×10^{-8}
	100	707,500	1.06×10^{2}	2.09×10^{4}	9.60×10^{-7}	$\mathbf{1.20 \times 10^{-8}}$
			7.12×10^{1}	1.93×10^{4}	3.97×10^{-7}	1.00×10^{-8}
			1.49×10^{2}	2.19×10^{4}	1.01×10^{-5}	1.70×10^{-8}

Table 6: Median, best and worst population-best objective function values (in first, second and third rows, respectively) obtained by three algorithms are shown for Set 1 multi-modal problems with one-sided initialization ($x_i \in [20, 60]$). For P8, we have used $x_i \in [10, 30]$, as within $x_i \in [20, 60]$, this problem is almost flat.

Prob.	n	FE	Parent-centric SBX	Mean-centric SBX	SA-SBX	M-SA-SBX
$P5$	20	22,500	8.42×10^{-7}	2.29	$\mathbf{4.00 \times 10^{-8}}$	9.70×10^{-8}
			1.46×10^{-7}	1.65	1.00×10^{-8}	1.00×10^{-8}
			7.40×10^{-3}	2.78	1.97×10^{-2}	3.43×10^{-2}
	50	88,750	8.76×10^{-4}	4.71	$\mathbf{1.60 \times 10^{-8}}$	6.30×10^{-8}
			4.98×10^{-4}	4.22	1.00×10^{-8}	2.20×10^{-8}
			1.20×10^{-3}	5.47	7.40×10^{-3}	1.23×10^{-2}
	100	280,000	1.21	1.15×10^{1}	$\mathbf{1.10 \times 10^{-8}}$	5.58×10^{-7}
			1.16	1.06×10^{1}	1.00×10^{-8}	2.75×10^{-7}
			1.30	1.24×10^{1}	9.90×10^{-3}	1.05×10^{-7}
$P6$	20	27,000	5.83×10^{-7}	1.64×10^{1}	$\mathbf{1.10 \times 10^{-8}}$	1.50×10^{-8}
			3.14×10^{-7}	1.01×10^{1}	1.00×10^{-8}	1.00×10^{-8}
			9.65×10^{-7}	2.30×10^{1}	8.60×10^{-8}	3.40×10^{-8}
	50	107,500	6.22×10^{-5}	1.96×10^{1}	$\mathbf{1.30 \times 10^{-8}}$	4.62×10^{-7}
			5.11×10^{-5}	1.55×10^{1}	1.00×10^{-8}	2.95×10^{-8}
			6.22×10^{-5}	1.97×10^{1}	3.90×10^{-8}	8.43×10^{-7}
	100	330,000	2.07×10^{-2}	2.49×10^{1}	$\mathbf{1.00 \times 10^{-8}}$	6.26×10^{-6}
			1.50×10^{-2}	2.30×10^{1}	1.00×10^{-8}	3.91×10^{-6}
			2.40×10^{-2}	2.82×10^{1}	1.80×10^{-8}	9.31×10^{-6}
$P7$	20	60,000	7.95	4.88×10^{3}	7.06×10^{1}	**5.97**
			3.98	3.22×10^{3}	3.98×10^{1}	9.95×10^{-1}
			1.19×10^{1}	7.86×10^{3}	5.22×10^{2}	1.19×10^{1}
	50	202,500	1.25×10^{1}	1.57×10^{4}	3.28×10^{1}	**3.98**
			2.47	1.29×10^{4}	2.29×10^{1}	9.95×10^{-1}
			2.82×10^{2}	1.89×10^{4}	5.87×10^{1}	1.09×10^{1}
	100	495,000	1.23×10^{3}	4.24×10^{4}	5.37×10^{1}	**6.99**
			1.17×10^{3}	3.63×10^{4}	4.57×10^{1}	1.47
			1.27×10^{3}	4.72×10^{4}	7.16×10^{1}	1.19×10^{1}
$P8$	20	33,000	2.92×10^{-6}	1.93×10^{1}	4.72×10^{-7}	$\mathbf{3.40 \times 10^{-8}}$
			1.51×10^{-6}	1.92×10^{1}	1.04×10^{-7}	1.30×10^{-8}
			8.48×10^{-6}	1.95×10^{1}	2.17	1.06×10^{-7}
	50	150,000	1.71×10^{-4}	1.94×10^{1}	$\mathbf{1.20 \times 10^{-8}}$	5.60×10^{-8}
			1.23×10^{-4}	1.94×10^{1}	1.00×10^{-8}	3.60×10^{-8}
			2.80×10^{-4}	1.95×10^{1}	2.70×10^{-8}	9.30×10^{-8}
	100	465,000	3.41	1.96×10^{1}	$\mathbf{1.50 \times 10^{-8}}$	5.07×10^{-7}
			2.77	1.95×10^{1}	1.00×10^{-8}	3.75×10^{-7}
			4.27	1.96×10^{1}	2.10×10^{-8}	7.48×10^{-7}

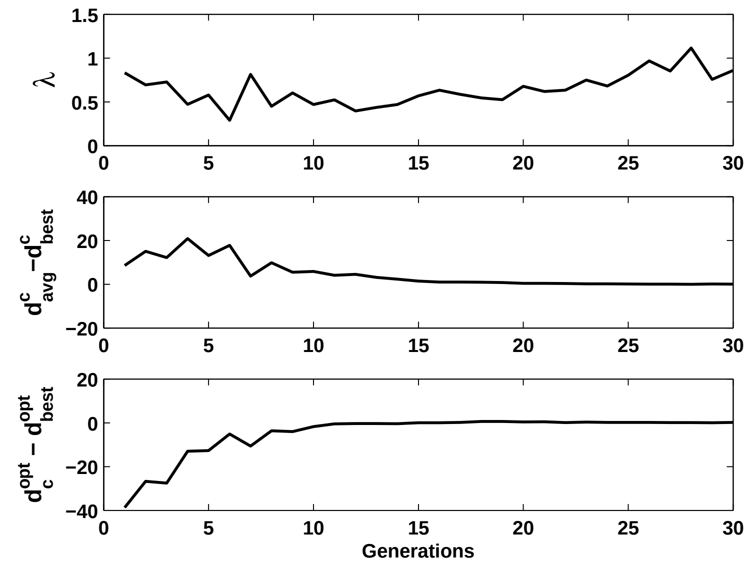


Figure 14: Variation of λ for 20-variable problem P1 with symmetric initialization.

finds it to be better to reduce λ to enable convergence to the optimum.

5.2 Results on Set 2 Test Problems

These problems are supposedly more difficult to solve than Set 1 test problems due to the biasness and rotational linkage associated with the variables. Table 7 shows median, best and worst objective function value in 20 different runs of the algorithms. Even in these problems, the proposed SA-SBX procedure performs better than RGAs with parent-centric and mean-centric operators alone. The proposed self-adaptation procedure seems to provide a reasonable extent of offspring solutions compared to the parent solutions to navigate close to the true optimal solutions. Figures 18(a) and (b) show the convergence pattern of three algorithms on problems P10 and P13. SA-SBX performs better than the two RGAs.

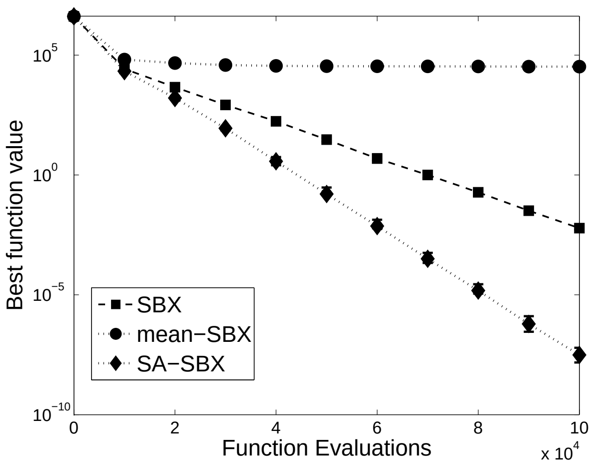
6 Modified SA-SBX (M-SA-SBX) Operator

In the SA-SBX approach suggested above, the parameter λ is calculated once at every generation and is kept fixed for all recombinations and for all variables in the particular generation. But in a large dimensional problem, it is likely that for some variables the population-best may lie at the edge of the population and in certain variables the population-best may lie well inside the population. We illustrate this phenomenon in Figure 19 which shows an intermediate GA population. It is clear that for the population, the parameter λ is greater than one, as the d_{best} is greater than d_{avg}. This means that with the SA-SBX operator a parent-centric operation will be performed on both variables x_1 and x_2. But clearly the figure depicts that for an effective search, the variable x_1 must be operated with a primarily mean-centric recombination operator and variable x_2 with a parent-centric recombination operator. Importantly, this difficulty will get worse in dealing with a large-dimensional problem. Thus there is a need for a variable-wise λ_i parameter, instead of a single λ parameter to improve the performance of the proposed SA-SBX operator.

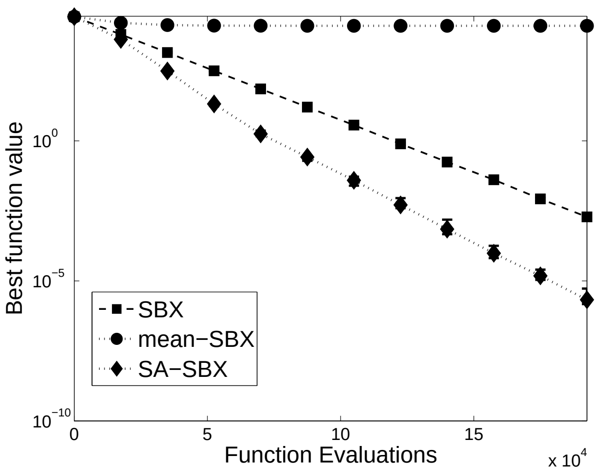
We argue a step further here and suggest that ideally the decision to perform a parent-centric or a mean-centric operation must not only depend on whether the population-best solution is at the edge of the population or inside variable-wise, but also must depend on location of the participating

Table 7: Objective function value (median of 20 runs) for all three algorithms are shown for Set 2 problems. Initialization ranges are mentioned in Table 2.

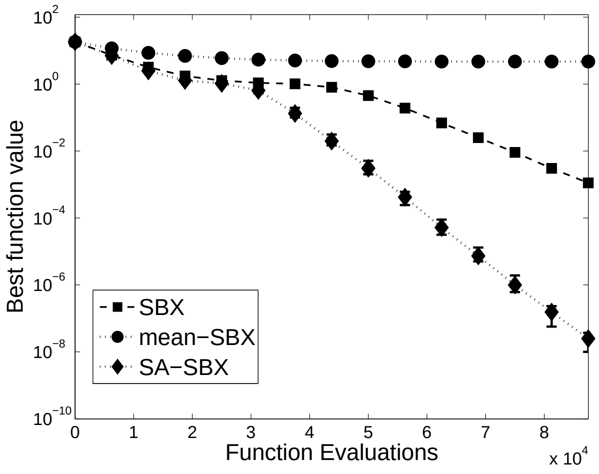
Prob.	n	FE	Parent-centric SBX	Mean-centric SBX	SA-SBX	M-SA-SBX
$P9$	10	6,500	3.53×10^{-5}	2.35×10^{2}	2.62×10^{-2}	$\mathbf{6.60 \times 10^{-8}}$
			5.30×10^{-6}	5.77×10^{-1}	3.56×10^{-7}	1.00×10^{-8}
			1.37×10^{-1}	1.43×10^{3}	2.42×10^{2}	2.31×10^{-6}
	30	42,000	7.32×10^{-5}	1.22×10^{3}	$\mathbf{1.00 \times 10^{-8}}$	3.20×10^{-8}
			3.08×10^{-5}	6.17×10^{2}	1.00×10^{-8}	1.00×10^{-8}
			1.27×10^{-4}	2.51×10^{3}	6.60×10^{-8}	1.62×10^{-7}
	50	95,000	1.95×10^{-2}	3.94×10^{3}	$\mathbf{1.00 \times 10^{-8}}$	3.72×10^{-6}
			1.07×10^{-2}	2.35×10^{3}	1.00×10^{-8}	2.10×10^{-6}
			3.02×10^{-2}	5.47×10^{3}	5.20×10^{-8}	7.24×10^{-6}
$P10$	10	119,500	3.34	5.67×10^{3}	2.66×10^{-4}	$\mathbf{8.30 \times 10^{-8}}$
			4.16×10^{-4}	3.39×10^{3}	1.67×10^{-5}	1.00×10^{-8}
			2.72×10^{2}	8.90×10^{3}	6.61×10^{1}	1.18×10^{-6}
	30	885,000	4.70×10^{-3}	2.14×10^{4}	9.72×10^{-4}	$\mathbf{2.40 \times 10^{-8}}$
			3.17×10^{-4}	1.65×10^{4}	5.90×10^{-4}	1.00×10^{-8}
			1.41×10^{-2}	2.76×10^{4}	2.70×10^{-3}	1.94×10^{-7}
	50	1,000,000	3.71×10^{5}	3.99×10^{4}	1.85×10^{2}	$\mathbf{6.80 \times 10^{-2}}$
			3.71×10^{5}	3.48×10^{4}	9.38×10^{1}	6.80×10^{-2}
			3.71×10^{5}	4.73×10^{4}	2.74×10^{2}	2.17×10^{-1}
$P11$	10	15,500	1.25×10^{-1}	1.21×10^{3}	6.69×10^{-2}	$\mathbf{5.70 \times 10^{-8}}$
			2.52×10^{-4}	9.23×10^{1}	2.50×10^{-3}	1.00×10^{-8}
			7.28	3.81×10^{3}	3.55×10^{2}	1.07×10^{-4}
	30	324,000	3.22×10^{-5}	2.26×10^{4}	2.77×10^{-5}	$\mathbf{4.10 \times 10^{-8}}$
			5.60×10^{-8}	1.16×10^{4}	8.17×10^{-7}	1.00×10^{-8}
			1.66×10^{-4}	4.00×10^{4}	7.32×10^{-4}	1.25×10^{-5}
	50	1,000,000	2.58×10^{-4}	6.85×10^{4}	1.60×10^{-3}	$\mathbf{9.89 \times 10^{-6}}$
			4.98×10^{-6}	3.89×10^{4}	1.03×10^{-5}	2.18×10^{-7}
			1.70×10^{-3}	1.05×10^{5}	5.50×10^{-3}	2.85×10^{-5}
$P12$	10	1,000,000	5.75×10^{1}	1.05×10^{7}	2.97×10^{2}	**3.98**
			5.91×10^{1}	4.05×10^{5}	1.54×10^{-2}	6.14×10^{-5}
			9.11×10^{3}	5.83×10^{7}	1.68×10^{6}	4.76×10^{3}
	30	1,000,000	1.04×10^{2}	9.26×10^{7}	6.88×10^{1}	$\mathbf{1.74 \times 10^{1}}$
			4.96	2.93×10^{7}	4.19	1.72×10^{-1}
			1.63×10^{4}	2.58×10^{8}	1.43×10^{3}	1.27×10^{3}
	50	1,000,000	1.87×10^{2}	3.47×10^{8}	1.13×10^{2}	$\mathbf{4.19 \times 10^{1}}$
			4.25×10^{1}	2.22×10^{8}	4.46	4.13×10^{1}
			2.41×10^{4}	3.89×10^{8}	1.38×10^{3}	3.12×10^{3}
$P13$	10	32,000	$\mathbf{9.26 \times 10^{-1}}$	1.43×10^{3}	1.01	4.57
			7.40×10^{-3}	8.09×10^{2}	3.29×10^{-2}	1.79×10^{-1}
			5.17	1.79×10^{3}	2.02×10^{2}	2.68×10^{1}
	30	226,500	1.05×10^{-2}	1.87×10^{3}	1.18×10^{-2}	$\mathbf{9.90 \times 10^{-3}}$
			7.70×10^{-3}	1.50×10^{3}	9.90×10^{-3}	7.50×10^{-3}
			2.50×10^{-2}	2.41×10^{3}	1.35×10^{-2}	3.90×10^{-2}
	50	685,000	4.12×10^{-6}	3.28×10^{3}	2.34×10^{-5}	$\mathbf{3.30 \times 10^{-8}}$
			2.14×10^{-6}	2.80×10^{3}	1.00×10^{-5}	1.00×10^{-8}
			1.23×10^{-2}	4.25×10^{3}	1.23×10^{-2}	1.48×10^{-2}
$P14$	10	992,000	21.05	1.61×10^{3}	6.35	**1.17**
			3.14×10^{-1}	1.07×10^{2}	1.00×10^{-8}	4.02×10^{-6}
			2.63×10^{3}	22 9.72×10^{3}	1.96×10^{4}	6.85×10^{4}



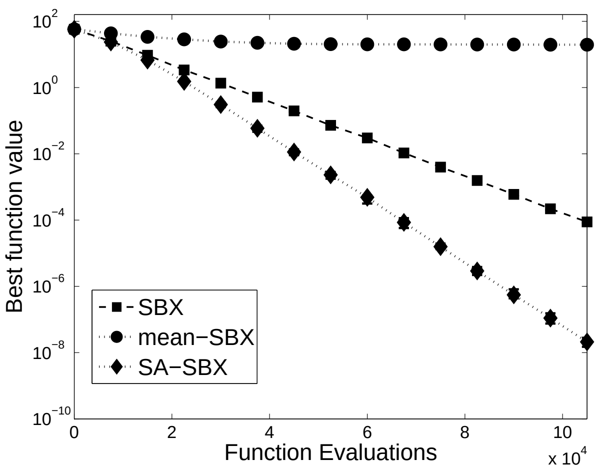
(a) P2



(b) P4



(c) P5

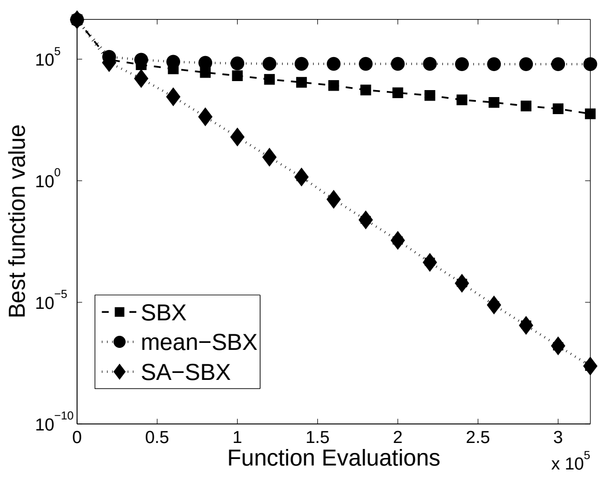


(d) P6

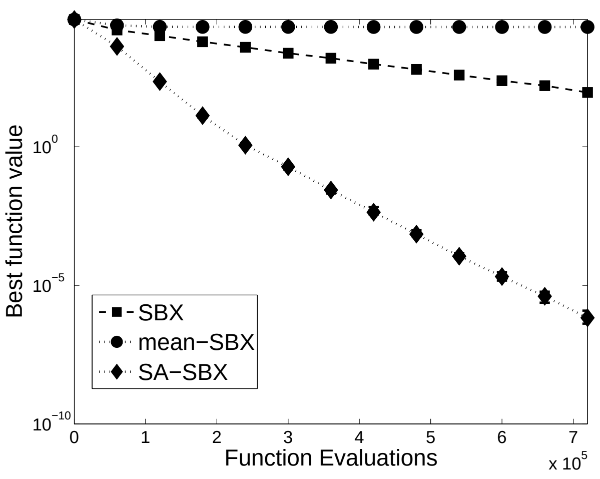
Figure 15: Population-best f for 50-variable problems P2, P4, P5 and P6 obtained using RGAs with SA-SBX, original parent-centric SBX, and mean-centric operator initialized on one side of the optimum.

parent solutions compared to the population-best solution in the search space. For this purpose, we suggest a modified SA-SBX (or M-SA-SBX) procedure in which the parameter λ is calculated for every pair of participating parent solutions in the particular variable space under consideration. Let us consider Figure 20, in which points A and B are considered for a recombination operation. The definition of the λ parameter for each variable is defined in a slightly different way. First, the centroid of the two parent values in the particular dimension is computed. Then, the distance of the population-best variable value to the centroid is called as d_{best} and half the distance between parent values in the dimension is called d_{avg}. Thereafter, the λ for that variable is computed as before $\lambda = d_{\text{best}}/d_{\text{avg}}$. The figure illustrates these computations for both x_1 and x_2 variables. Points G1 and G2 are the corresponding centroids. It can be seen that for the pair (A and B) of parent solutions, variable x_1 will be treated as a mean-centric as $\lambda < 1$ and variable x_2 will be treated as a parent-centric operator due to λ being greater than 1. It is interesting to realize how these recombination operations will emphasize creation of offspring solutions below the two parent solutions on x_2 and within the parents in x_1 so as to create points close to the true optimum point.

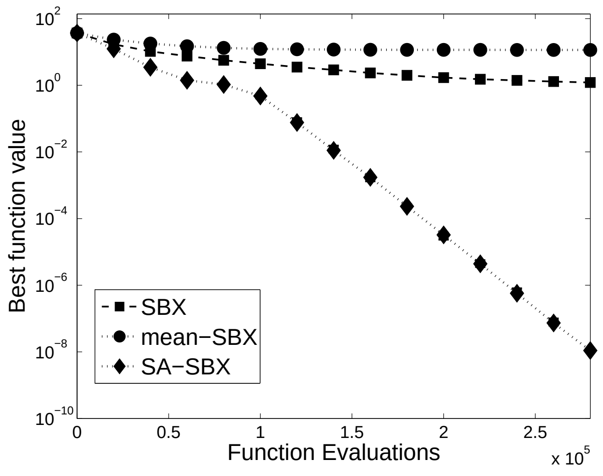
It is clear from the above discussion that the location of the population-best solution makes a big impact on the offspring population. Thus, the reliability of the proposed procedure to find the global optimum of a problem largely depends on the proximity of the current population-best



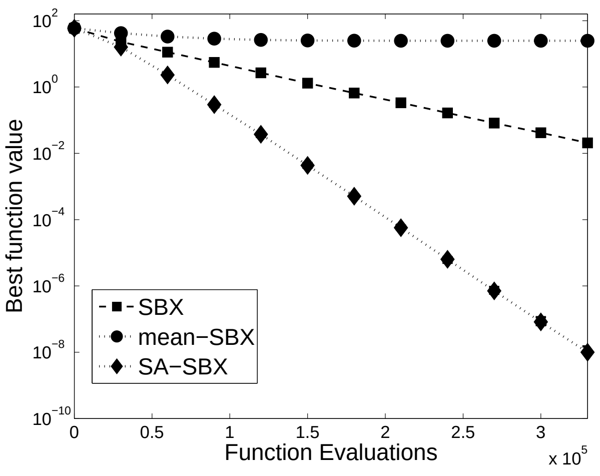
(a) P2



(b) P4



(c) P5



(d) P6

Figure 16: Population-best f for 100-variable problems P2, P4, P5 and P6 obtained using RGAs with SA-SBX, original parent-centric SBX, and mean-centric operator initialized on one side of the optimum.

solution with the global optimal solution. To get a reliable estimate of the current best solution, instead of simply taking the population-best solution, we identify top 2% population members and the centroid of these elite population members is considered as the population-best point in the recombination operation.

7 Results Using M-SA-SBX Operator

Here too, we do not use the mutation operator in order to evaluate the performance of the proposed recombination operator alone. We use identical parameter settings as before. No mutation operator is used to investigate the the performance of the proposed recombination operator alone.

7.1 Understanding M-SA-SBX Operator

First, we consider a single-variable optimization problem to demonstrate how the redefined parameter λ changes with generation. We consider the objective function $f(x) = x^2$ and initialize a GA population randomly within $x \in [80, 100]$, knowing that the optimum lies at $x^* = 0$ with $f^* = 0$. Thus, in this case, the GA population has to first approach the optimum and then focus in

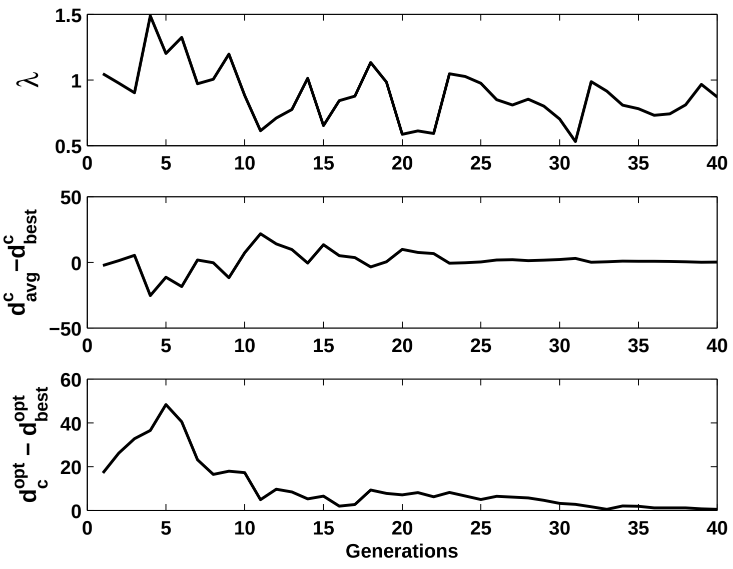
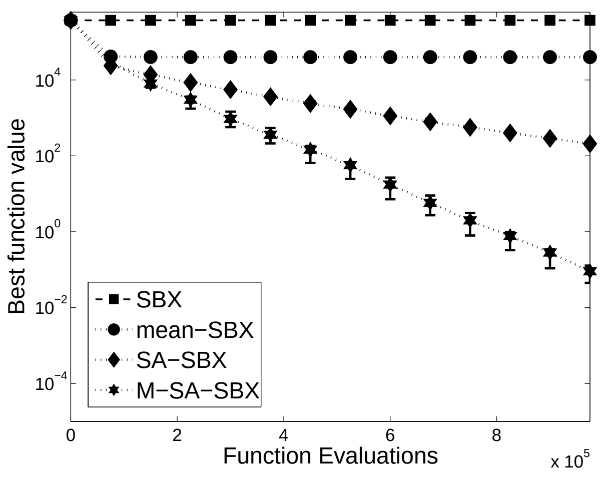


Figure 17: Variation of λ for 20-variable problem P1 with one-sided initialization.

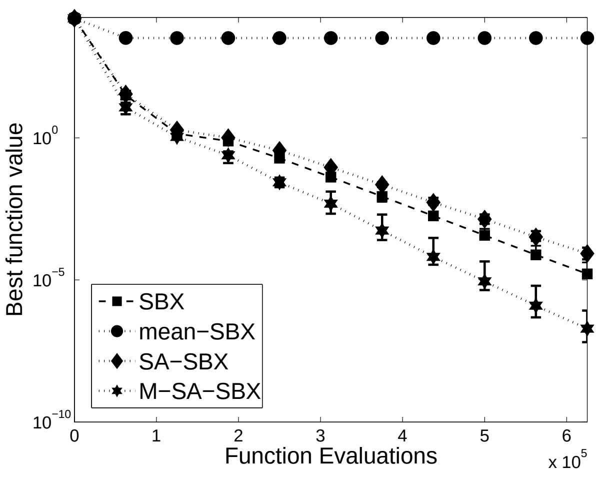
finding the true optimum solution. We use a population of 20 and run RGA for a few generations. No mutation operator is used. Other parameters are the same as before. To compare the performance of our proposed RGA with M-SA-SBX operator, we also run RGA with the original SBX operator on the same problem with identical initial populations. Figure 21 shows the variation of the population-best objective function value with generation counter. It can be seen that clearly the proposed approach is much quicker in reducing the objective value close to the true optimum objective value of $f^* = 0$. With the M-SA-SBX approach, every recombination operation is characterized with a different λ value. To understand the self-adaptive variation of this parameter, we identify the median λ at each generation and plot it with the generation counter in Figure 22. Although the λ value is not used in the original SBX operator, we still compute the median value of λ for the SBX recombinations at every generation and plot it in the figure. First, both approaches maintain a λ value higher than one, meaning that the population-best solution is at the edge of the population in the initial generations of the simulation. Since the initial population is created far away from the true optimum, the population must approach the optimal region in the initial generations and this performance is desired. But it can be seen from the figure that RGA with the original SBX operator keeps the population-best solution far away from the bulk of the population. Since the SBX operator creates 50% offspring solutions inside the region bounded by the parents and 50% outside, there is a relatively small probability for creating offspring solutions away from the population and towards the optimum. However, in the M-SA-SBX operator most recombinations produce offspring solutions outside the range of parents in the early generations on this problem, and the population-best solution and the bulk of the population move together towards the optimum solution. The median λ value is much smaller than that in the SBX run. The move of the population and the population-best together helps the modified approach to reach the true optimum faster.

7.2 Results on Test Problems

Having understood the difference in the performance of the M-SA-SBX with respect to the original SBX operator, we now present the results on the test problems of this study. The final column of Tables 3 and 4 shows the performance of this modified self-adaptive recombination operator on Set 1



(a) P10



(b) P13

Figure 18: Population-best objective function value versus generation counter for 50-variable version of problems P10 and P13 using SA-SBX, SBX and mean-centric SBX operators. Performance of M-SA-SBX is also shown.

problems initialized symmetrically around the optimum. It is clear that the RGA with M-SA-SBX has outperformed RGAs with previous SA-SBX, purely parent-centric and purely mean-centric recombination operators. The flexibility to choose different λ values for different variables and participating parents allow a more efficient search than simply marking all recombination events in a generation of one type or another.

A similar superior performance of M-SA-SBX operator can also be observed from Tables 5 and 6, in which Set 1 problems are solved with an one-sided initialization procedure. Finally, the multi-modal problems are also solved better with M-SA-SBX operator.

7.3 Defining Population-best Point

In the above simulations, we have defined a population-best point that is the centroid of top $\zeta = 2\%$ of the population members. Before we recommend this number, in this subsection, we perform a parametric study to obtain an idea of a reasonable value for this parameter.

We consider four test problems – two unimodal (P1 and P9) and two multi-modal (P6 and P13) – for this purpose. We consider symmetric initialization process and perform 20 runs with four different ζ values: 1, 2, 5 and 10. The population is sorted in ascending oder of the objective function value and the centroid of the top $\zeta\%$ population members is used as the population-best point in the M-SA-SBX operator. Table 8 shows the results on four test problems. It is clear that around $\zeta = 2\%$ performs the best. The results indicate that making M-SA-SBX dependent only on the population-best solution (second column) may not prove that fruitful because for a multi-dimensional problem it may not always be true that all variables for the population-best solution are closer to true optimum compared to any other population member. It may very well happen that some of the variables are very close to the optimum (and hence the function value is small) while others are far from the optimum. In such a case, performing M-SA-SBX with the population-best solution will take the algorithm closer to the optimum in certain variables while in others it will not (or even may take away from the optimum). However, the centroid of the top $\zeta\%$ population members as the best point to compute λ values may average out some of the isolated properties of the singleton population-best solution and this may prevent the algorithm from going in a wrong direction. This is what is observed in Table 8 with larger ζ values. However, if a large ζ value is chosen, the concept of a good solution in the computation of λ value gets diluted and best point

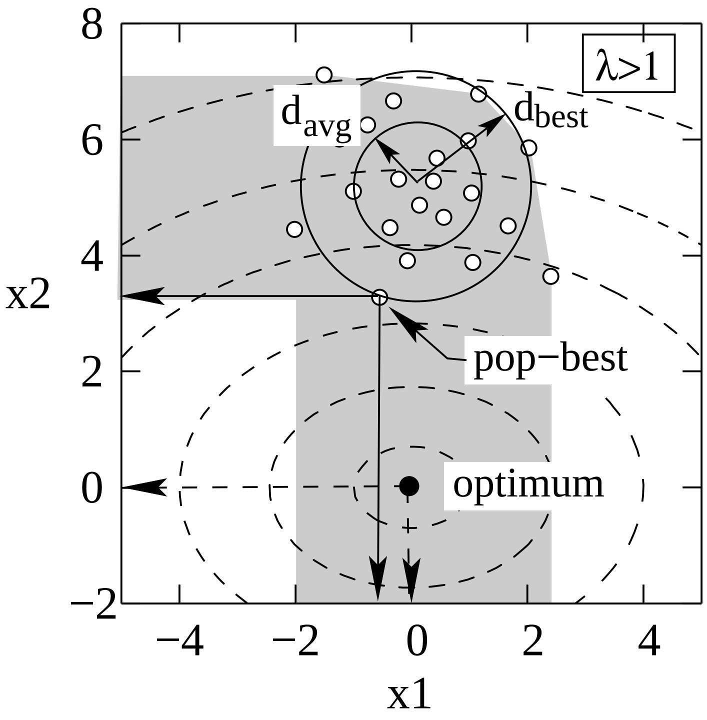


Figure 19: On x_1-variable a mean-centric is desired, on the other hand, on x_2 variable a parent-centric recombination is desired.

comes closer to the centroid of the population in most situations. We observe the deterioration in performance with $\zeta = 10\%$ already in the table. In the case of multi-modal problems, basing M-SA-SBX only on the current population-best solution (which may not be the global optimum) may eventually lead the search to have a premature convergence to a local optimal solution. Based on the performance on all four problems (unimodal and multi-modal), a value of $\zeta = 2\%$ seems to be good compromise.

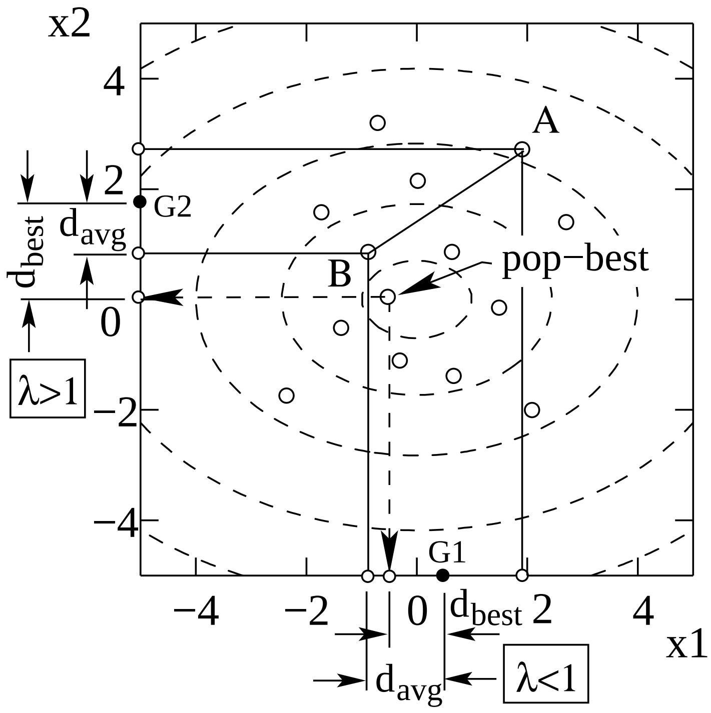
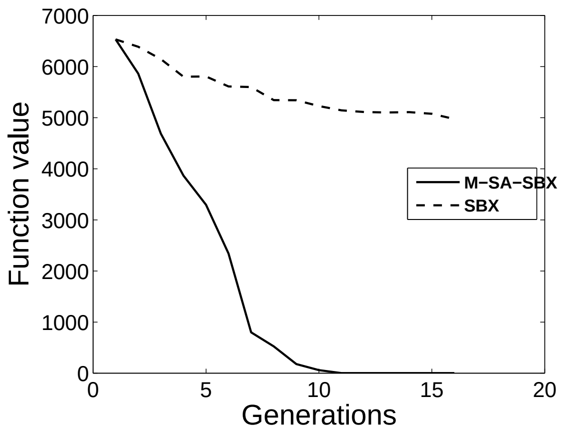


Figure 20: The computation of λ is redefined for every pair of recombination operation in each dimension.



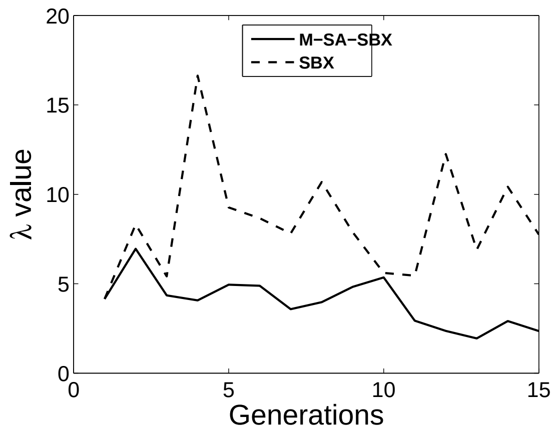


Figure 21: Population-best objective value reduces at a much faster rate with M-SA-SBX operator for the quadratic problem.

Figure 22: Median value of parameter λ for M-SA-SBX and SBX operators for the quadratic problem.

Table 8: Best objective function value (median of 20 runs) obtained for different proportions (ζ) of top individuals used to compute the population-best point. In the first column, the quantity in brackets indicates the number of variables, the next row indicates initialization range and the third row indicates the total function evaluations allowed in a run.

Problem	Pop-best	1%	2%	5%	10%
$P1$ (50)	1.73×10^{6}	1.74×10^{-4}	$\mathbf{3.10 \times 10^{-8}}$	1.96×10^{-6}	2.92×10^{-4}
$[20, 60]^n$	$1.73 \times 10^{6})$	5.93×10^{-5}	1.40×10^{-8}	8.98×10^{-7}	1.11×10^{-4}
112,500	2.22×10^{6}	4.49×10^{-4}	6.80×10^{-8}	5.37×10^{-6}	4.56×10^{-4}
$P6$ (50)	3.73×10^{3}	1.33×10^{-5}	1.90×10^{-8}	$\mathbf{1.00 \times 10^{-8}}$	4.62×10^{-7}
$[20, 60]^n$	1.19×10^{3}	5.95×10^{-6}	1.00×10^{-8}	1.00×10^{-8}	2.05×10^{-7}
127,500	4.24×10^{4}	2.11×10^{-5}	3.20×10^{-8}	1.20×10^{-8}	1.07×10^{-6}
$P9$ (30)	4.02×10^{1}	8.56×10^{-7}	$\mathbf{1.00 \times 10^{-8}}$	3.80×10^{-8}	6.64×10^{-6}
$[-100, 100]^n$	6.45	3.57×10^{-7}	1.00×10^{-8}	1.00×10^{-8}	2.36×10^{-6}
45,000	2.26×10^{2}	6.27×10^{-6}	1.70×10^{-8}	1.13×10^{-7}	2.20×10^{-5}
$P13$ (30)	1.01	$\mathbf{9.90 \times 10^{-3}}$	$\mathbf{9.90 \times 10^{-3}}$	3.81×10^{-2}	1.49×10^{-1}
$[0, 600]^n$	1.91×10^{-1}	7.40×10^{-3}	7.50×10^{-3}	1.90×10^{-2}	7.10×10^{-2}
226,500	1.40	2.96×10^{-2}	3.90×10^{-2}	5.76×10^{-2}	2.00×10^{-1}

8 Modified Self-Adaptive SBX for Multi-objective Optimization

Having shown the effectiveness of the proposed M-SA-SBX operator in single-objective unimodal and multi-modal problems, we now investigate its usefulness in solving multi-objective optimization problems for finding a set of Pareto-optimal solutions [1].

One important aspect of the M-SA-SBX approach is its use of the population-best solution and unfortunately in multi-objective optimization problems there is no single population-best solution. The presence of multiple conflicting objectives causes the set of non-dominated solutions in the current population to be the population-best solutions. Different evolutionary multi-objective optimization (EMO) algorithms define and treat non-dominated solutions differently [3, 19, 18]. Here, we use the elitist non-dominated sorting GA or NSGA-II [3] as our base multi-objective optimization algorithm. The recombination operation is redefined as follows.

Note that for the parent population P_t at generation t, the non-dominated set $\mathcal{F}_1$ is already known. For a recombination operation, we identify two parent solutions as usual using the tournament selection operator that prefers a non-dominated solution over a dominated solution and a less-crowded solution when two solutions being compared belong to an identical front. For the M-SA-SBX recombination of two parent solutions, we tried a number of schemes to define the population-best solution from the $\mathcal{F}_1$ set. The solution in $\mathcal{F}_1$ that is closest to the centroid of the two parent solutions in the decision-variable space has been tried as a viable approach. The distance has been computed in the objective space as well as another viable approach. Both these considerations have resulted in a quick loss of diversity of the population and the resulting algorithm did not perform well. After a few other implementations, we have observed that a strategy in which a random solution from $\mathcal{F}_1$ is picked as the population-best solution wins over all strategies tried in this study. By choosing a random non-dominated solution as population-best solution for M-SA-SBX operation makes a balance between the much-needed diversity on the obtained solutions and simultaneously creating better non-dominated solutions. The rest of the NSGA-II procedure [3] is kept the same.

8.1 Results on Test Problems

To evaluate the performance of the proposed multi-objective implementation of the M-SA-SBX operator, we apply the procedure to standard two-objective ZDT and three-objective DTLZ problems [5]. The problems used in this study are presented in Table 9. For comparing the performance of NSGA-II with M-SA-SBX, we use the NSGA-II procedure with original SBX operator. The hypervolume measure [20] of the obtained front is used as a performance indicator. In each case, 20 runs are made from different initial populations until a pre-specified value of hypervolume is attained. The number of generations required to attain the specific hypervolume value is then presented. In all cases, a population of size 100, a crossover probability of $p_c = 0.9$, a polynomial mutation with a probability of $p_m = \min(1/n, 1/30)$ (where n is the number of variables in the problem) and $\eta_c = 2$ and $\eta_m = 20$ are followed. For these problems, we follow the standard initialization procedure given in the problem formulation and is also mentioned in Table 9.

Table 10 tabulates the best, median and worst number of generations needed to achieve a particular hypervolume measure (HV) specified in the table. It is clear that the M-SA-SBX makes the overall NSGA-II procedure faster to arrive at an identical HV value to that with the SBX procedure. The hypervolume measure depends on how close the obtained front is the true front and also how diverse a set of points are achieved. Due to the creation of appropriate offspring solutions by M-SA-SBX in a self-adaptive manner, we observe a faster and more efficient performance of the proposed procedure.

To demonstrate the similarity in the working of the proposed procedure with that of the NSGA-II and SBX operator, we now plot the obtained trade-off fronts for both procedures from a typical simulation run. Figures 23(a) and (b) show that on ZDT1 problem both procedures have performed

Table 9: A list of multi-objective test problems used in the study is presented. All objective functions are to be minimized.

Prob.	n	Variable bounds	Objective function
		Two-objective problems	
ZDT1	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{x_1/g(\mathbf{x})}]$ $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT2	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - (x_1/g(\mathbf{x}))^2]$ $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT3	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{x_1/g(\mathbf{x})} - \frac{x_1}{g(\mathbf{x})}\sin(10\pi x_1)]$ $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT4	10	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2, .., n$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{x_1/g(\mathbf{x})}]$ $g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^{n} [x_i^2 - 10\cos(4\pi x_i)]$
ZDT6	10	$[0, 1]$	$f_1(\mathbf{x}) = 1 - \exp(-4x_1)\sin^6(4\pi x_1)$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2]$ $g(\mathbf{x}) = 1 + 9[(\sum_{i=2}^{n} x_i)/(n-1)]^{0.25}$
		Three-objective Problems	
DTLZ1	7	$[0, 1]$	$f_1(\mathbf{x}) = \frac{1}{2}x_1 x_2(1 + g(\mathbf{x}))$ $f_2(\mathbf{x}) = \frac{1}{2}x_1(1 - x_2)(1 + g(\mathbf{x}))$ $f_3(\mathbf{x}) = (1 - x_1)g(\mathbf{x})$ $g(\mathbf{x}) = 100\sum_{i=3}^{n}\left[1 + (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right]$
DTLZ2	12	$[0, 1]$	$f_1(\mathbf{x}) = (1 + g(\mathbf{x}))\cos(x_1\pi/2)\cos(x_2\pi/2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x}))\cos(x_1\pi/2)\sin(x_2\pi/2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x}))\sin(x_1\pi/2)$ $g(\mathbf{x}) = \sum_{i=3}^{n}(x_i - 0.5)^2$

almost equally well, except that the hypervolume plot shows a slightly quicker performance of M-SA-SBX approach. The minimum and maximum HV values achieved in 20 runs are also marked in hypervolume plots.

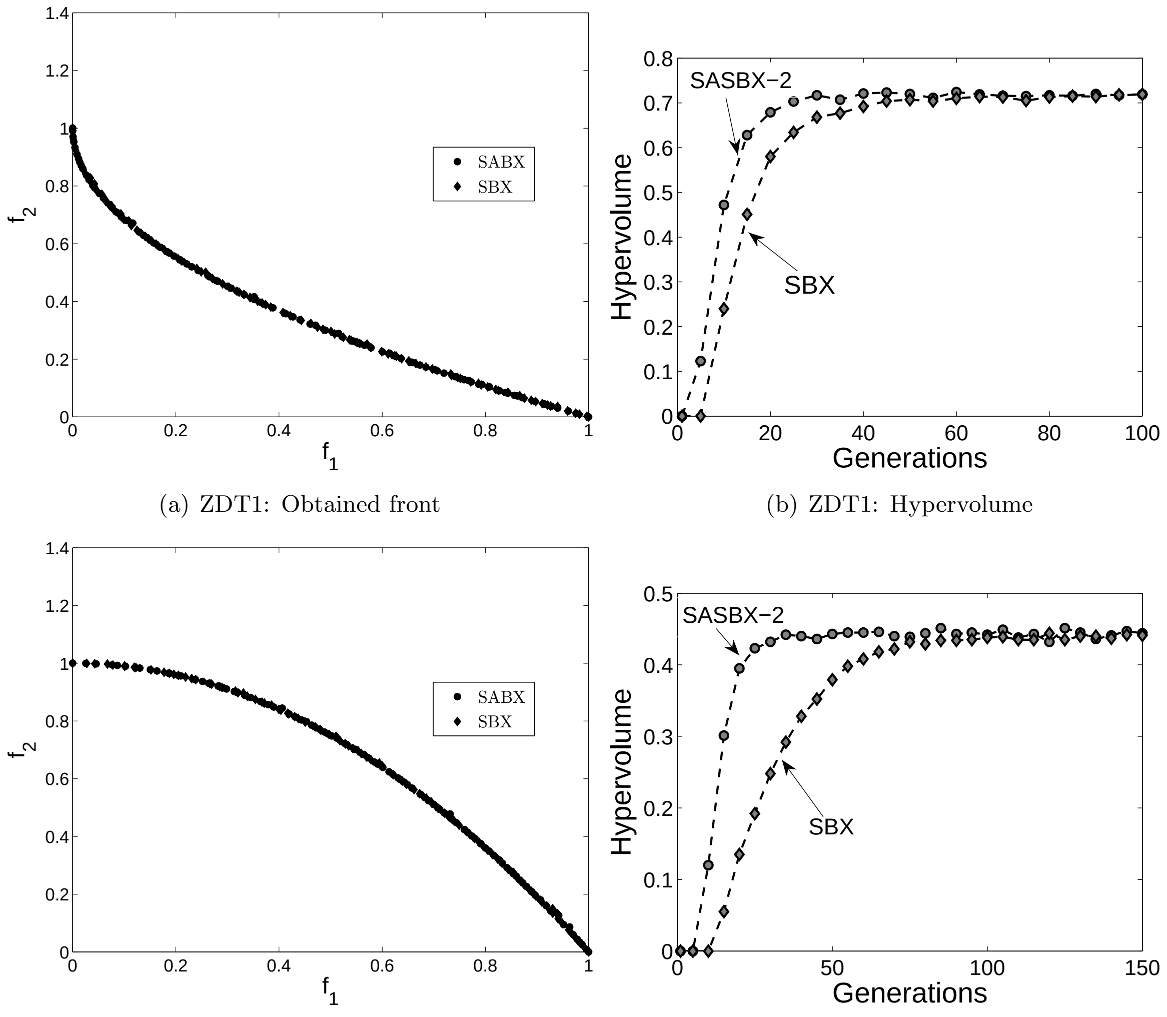
On ZDT2, a similar observation can be made from Figures 23(c) and (d). On ZDT3, the performance of both approaches are more or less same (Figure 24), but on ZDT4 NSGA-II with M-SA-SBX performs better.

The performance of NSGA-II with M-SA-SBX is visibly better than NSGA-II with SBX on ZDT6 problem (Figures 25(a) and (b)). After a stipulated number of generations, both approaches reach the same trade-off front, but NSGA-II with M-SA-SBX is much faster in this problem. In three-objective DTLZ1 problem, although there is an early improvement of the hypervolume measure, due to the multi-modality present in this problem, the hypervolume stagnates at an intermediate phase of the algorithm. However, eventually NSGA-II with M-SA-SBX is able to achieve an identical hypervolume measure to that of NSGA-II with the SBX operator. On DTLZ2 problem, both algorithms perform in a similar manner.

Based on the above multi-objective simulation results, it can be concluded that for difficult problems (particularly ZDT4, ZDT6 and DTLZ1) the M-SA-SBX recombination operator has demon-

Table 10: Number of generations required to reach the indicated hypervolume value (median of 20 runs) for both algorithms are shown. Smallest and highest number or generations are shown in second and third rows, respectively. 'Succ.' indicates the number of successful runs out of 20 runs.

Problem	Hypervolume	M-SA-SBX		SBX	
		# gen.	Succ.	# gen.	Succ.
$ZDT1$	0.721	**40**	14	90	15
		30		50	
		−−		−−	
$ZDT2$	0.442	**60**	20	105	20
		30		65	
		215		495	
$ZDT3$	0.830	**35**	20	60	20
		25		40	
		280		340	
$ZDT4$	0.720	**175**	16	300	13
		145		185	
		−−		−−	
$ZDT6$	0.390	**125**	20	220	20
		95		165	
		240		375	
$DTLZ1$	0.962	**215**	20	310	16
		195		300	
		240		−−	
$DTLZ2$	0.488	65	18	**55**	17
		45		35	
		−−		−−	



(a) ZDT1: Obtained front (b) ZDT1: Hypervolume

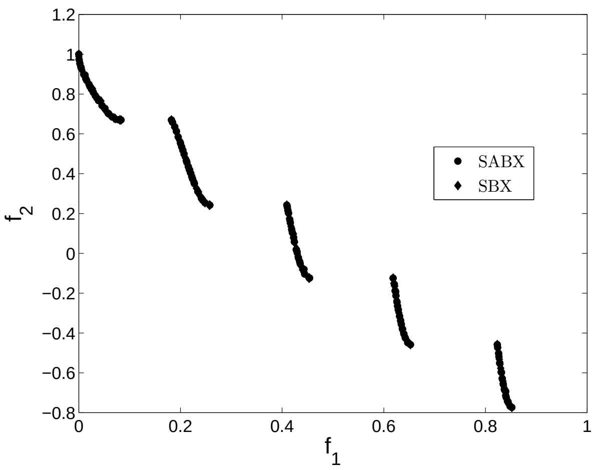
(c) ZDT2: Obtained front (d) ZDT2: Hypervolume

Figure 23: Typical trade-off fronts and corresponding variation of hypervolume measures over 20 runs for ZDT1 and ZDT2 problems using NSGA-II with M-SA-SBX and NSGA-II with original SBX.

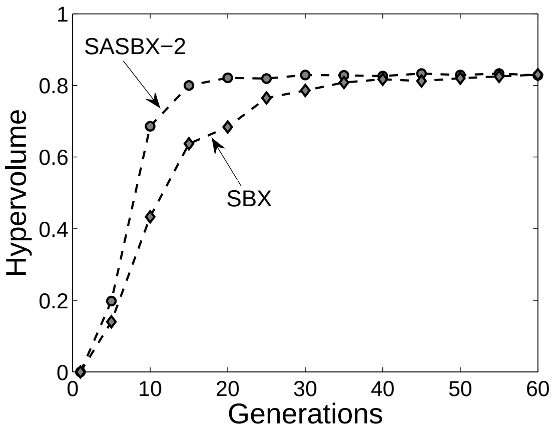
strated an edge over the SBX operator and in simpler problems both procedures perform in a more or less identical manner. The use of the M-SA-SBX operator by choosing a random current non-dominated solution as the population-best solution for λ computation is a simple concept that seems to have worked better or similar to existing SBX operator. This study encourages development of further more specific approaches for computing λ values for multi-objective optimization.

9 Conclusions

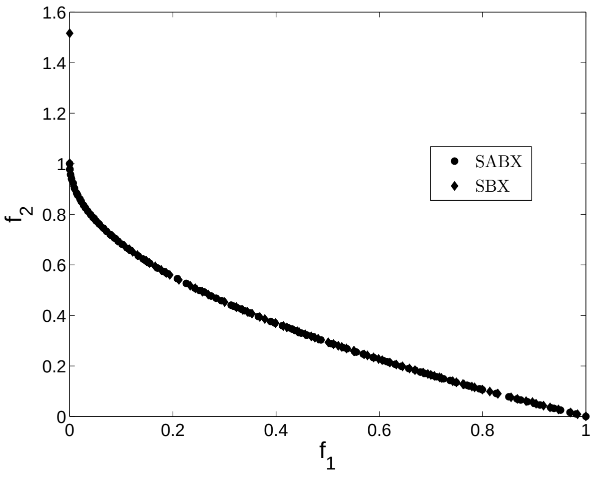
Real-parameter genetic algorithms (RGAs) have become popular after the advent of blending operations used as a recombination operator between two real-valued vectors. Despite specific real-parameter evolutionary algorithms, such as evolution strategies, differential evolution, particle swarm optimization, RGAs have remained as viable and competing algorithms for real-parameter optimization. In this paper, we have suggested a self-adaptive version of a real-parameter recombination operator (the simulated binary crossover or SBX) by using the current population statistics. The location of parents in a SBX has been decided by computing virtual parents that is controlled by the relative location of the population-best solution compared to the bulk of the population. A



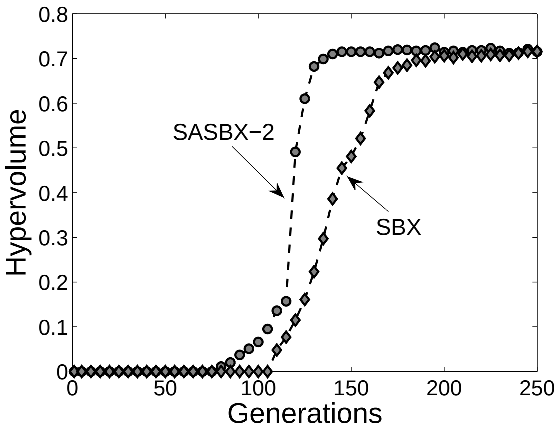
(a) ZDT3: Pareto Front



(b) ZDT3: Hypervolume



(c) ZDT4: Pareto Front



(d) ZDT4: Hypervolume

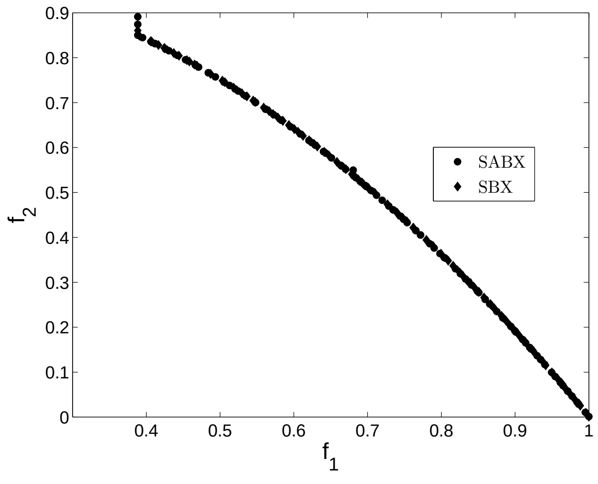
Figure 24: Typical trade-off fronts and corresponding variation of hypervolume measures over 20 runs for ZDT3 and ZDT4 problems using NSGA-II with M-SA-SBX and NSGA-II with original SBX.

generational RGA with the resulting SA-SBX operator has been found to outperform RGAs with a parent-centric SBX and a mean-centric operator alone on a number of standard unimodal and multi-modal test problems with and without linkage. The proposed algorithm has also been found to be efficient in handling situations in which the initial population does not bracket the optimum.

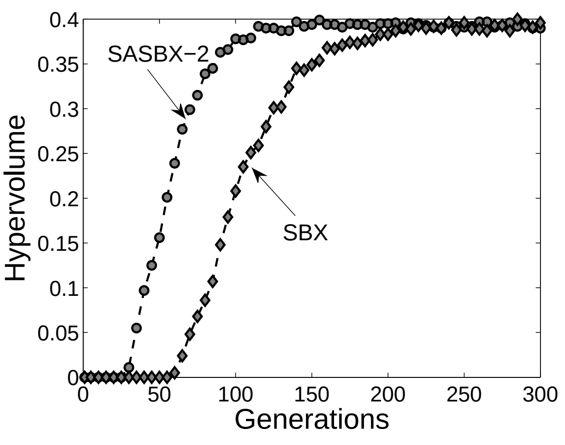
Based on a deeper understanding of the operator, we also suggest a modified SA-SBX operator that determines the extent of parent versus mean-centric operation for each variable and for each recombination. This allows different types (parent or mean-centric) of recombination for different variables on a pair of participating parents, thereby allowing the linkage among variables to be somewhat maintained from one generation to the next. Simulation results with M-SA-SBX operator have shown a better performance, in general, compared to the SA-SBX approach.

Finally, the modified SA-SBX operator has been implemented with NSGA-II to solve two and three-objective multi-objective optimization problems. The self-adaptive idea has produced better or equivalent results to the original NSGA-II procedure on a number of standard test problems.

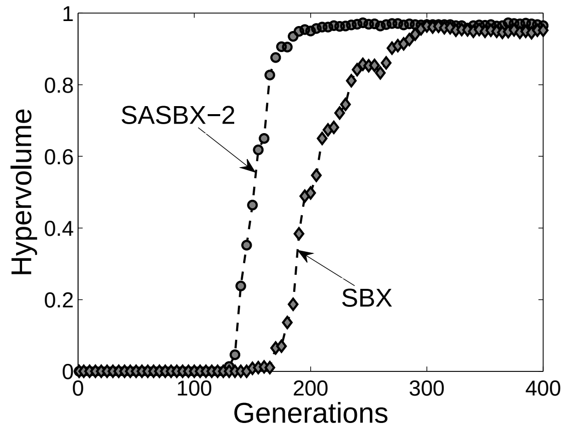
The purpose of this paper was not to compare the proposed algorithm with other evolutionary or classical optimization methods, but to investigate if a self-adaptive parent to mean-centric operation of the SBX recombination operator works better than the individual parent-centric or mean-centric SBX recombination operators alone. The population approach of evolutionary optimization



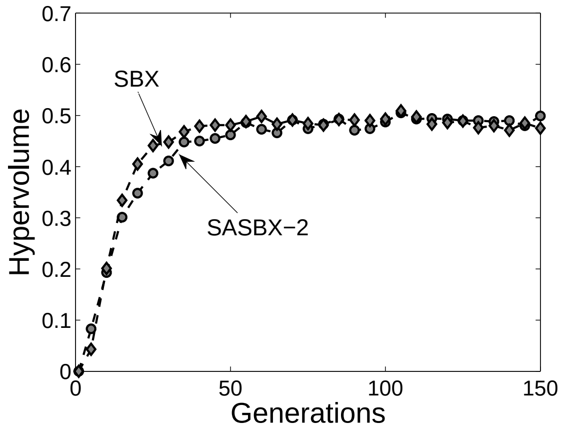
(a) ZDT6: Pareto Front



(b) ZDT6: Hypervolume



(c) DTLZ1: Hypervolume



(d) DTLZ2: Hypervolume

Figure 25: Typical trade-off fronts and corresponding variation of hypervolume measures over 20 runs for ZDT6, DTLZ1 and DTLZ2 problems using NSGA-II with M-SA-SBX and NSGA-II with original SBX.

algorithms has allowed a unique way to introduce self-adaptation of evolutionary operators. Self-adaptation of algorithmic parameters and operations are crucial to negotiate complex objective landscapes that an efficient optimization algorithm must adhere to. An EA's population, at every generation, should provide adequate information about how and where to create the new population so as to have a better navigation through the search space and approach the optimum. In this paper, we have suggested a viable approach utilizing the location of a population's current best solution from the bulk of the population members and invoke a recombination strategy to create offspring solutions most appropriate to the scenario. Simulation results with a selection operator and such a recombination operator alone has supported our arguments on a number of standard test problems. The approach now must be tested on other more challenging test problems and practical problems with other genetic operators such as mutation and elite-preserving operators. The complete algorithm now must be compared with other state-of-the-art real-parameter optimization algorithms to investigate its future in the realm of real-parameter evolutionary algorithms.

Acknowledgments

Authors acknowledge the Academy of Finland grant under research project number 133387. Both authors acknowledge the support provided by Michigan State University during the course of this study.

References

[1] K. Deb. *Multi-objective optimization using evolutionary algorithms.* Wiley, Chichester, UK, 2001.

[2] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[4] K. Deb, A. Anand, and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation Journal*, 10(4):371–395, 2002.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization*, pages 105–145. London: Springer-Verlag, 2005.

[6] L. J. Eshelman and J. D. Schaffer. Real-coded genetic algorithms and interval-schemata. In *Foundations of Genetic Algorithms 2 (FOGA-2)*, pages 187–202, 1993.

[7] D. E. Goldberg. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5(2):139–168, 1991.

[8] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strageties: The covariance matrix adaptation. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 312–317, 1996.

[9] F. Herrera, M. Lozano, and J. L. Verdegay. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review*, 12(4):265–319, 1998.

[10] James Kennedy, Russell C. Eberhart, and Y. Shi. *Swarm intelligence.* Morgan Kaufmann, 2001.

[11] I. Ono and S. Kobayashi. A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. In *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA-7)*, pages 246–253, 1997.

[12] N. J. Radcliffe. Forma analysis and random respectful recombination. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 222–229, 1991.

[13] H.-P. Schwefel. *Evolution and Optimum Seeking.* New York: Wiley, 1995.

[14] R. Storn and K. Price. Differential evolution – A fast and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[15] S. Tsutsui, M. Yamamura, and T. Higuchi. Multi-parent recombination with simplex crossover in real-coded genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pages 657–664, 1999.

[16] H.-M. Voigt, H. Mühlenbein, and D. Cvetković. Fuzzy recombination for the Breeder Genetic Algorithm. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 104–111, 1995.

[17] A. Wright. Genetic algorithms for real parameter optimization. In *Foundations of Genetic Algorithms 1 (FOGA-1)*, pages 205–218, 1991.

[18] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731, 2007.

[19] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2001. International Center for Numerical Methods in Engineering (CIMNE).

[20] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms – A comparative case study. In *Parallel Problem Solving from Nature V (PPSN-V)*, pages 292–301, 1998.