Self-Adaptive Parent to Mean-Centric Recombination for Real-Parameter Optimization

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

KanGAL Report Number 2011001

Abstract

Most real-parameter genetic algorithms (RGAs) use a blending of participating parent solutions to create offspring solutions through its recombination operator. The blending operation creates solutions either around one of the parent solutions (having a parent-centric approach) or around the centroid of the parent solutions (having a mean-centric approach). In this paper, we argue that a self-adaptive approach in which a parent-centric or a mean-centric approach is adopted based on population statistics is a better procedure than either approach alone. We propose a self-adaptive simulated binary crossover (SA-SBX) approach for this purpose. On a suite of eight unimodal and multi-modal test problems, we demonstrate that a RGA with SA-SBX approach performs reliably and consistently better in locating the global optimum solution than the RGA with the original parent-centric SBX operator and the well-known CMA-ES approach.

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 are recombined independently with a probability to create a new value. The resulting offspring solution is then formed by concatenating the created values from recombinations or chosen from one of the parent solutions. This procedure does not honor the necessary *linkage* between different variables needed to solve a problem adequately and is most suited for variable-wise separable problems. Some examples of variable-wise recombination operators are blend crossover (BLX) [5], fuzzy recombination (FR) [9], simulated binary crossover (SBX) [1], and others. On the other hand, in the vector-wise recombination operators, 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 propagate and honor the linkage among variables. Some examples are UNDX operator [8], simplex crossover (SPX) [7], parent-centric crossover (PCX) [2], and others. It is worth mentioning here that the search operation of differential evolution (DE) and particle swarm optimization (PSO) use a vector-wise recombination operation

The recombination operators (variable or vector-wise) can also be classified into two functionally different classes based on the location of creating offspring solution vis-a-vis the location of parent solutions. An earlier study [2] 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 around the mean of the participating parent solutions. 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. FR, SBX, PCX are examples of such an operator. Since parent solutions are tested to be better by the preceding selection operator, 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 expected to be away from each other thereby making the mean of parent solutions to lie completely in a new region which may not have been tested for its 'goodness' by the selection operator.

In this paper, we argue and demonstrate that while a parent-centric operator may be judged to be better early on in a simulation, a mean-centric operator may be found to be beneficial when population members crowd around the optimum. Thus, instead of using a parent-centric or a mean-centric recombination operator throughout a simulation, a better strategy would be to adaptively choose a parent-centric or a mean-centric operator depending on whether the population is approaching or crowded around the optimum.

In the remainder of the paper, we overview the difference between parent and mean-centric operators and indicate some popular recombination operators. We then consider a specific parent-centric operator (SBX) and modify the operator so that at every generation it behaves like a parent-centric or a mean-centric operator depending on the population statistics. Thereafter, we demonstrate the efficacy of the proposed modified algorithm on a number of unimodal and multi-modal test problems borrowed from the literature. The results are compared with the original parent-centric SBX operator, a purely mean-centric operator, and the well-known CMA-ES algorithm [6]. The proposed approach is found to perform consistently better in most problems than other methods including CMA-ES. Moreover, the methodology is also found to be scalable in 20 to 100 variable versions of the test problems. With a mention of immediate future studies, conclusions of this paper is finally made.

2 Real-Parameter Recombination Operators

The main distinguishing aspect of a recombination operator compared to other search operations in an evolutionary algorithm (EA) is that more than one evolving solutions are utilized to create one or more new offspring solutions. Since an EA population evolves with the generation counter, the creation of offspring solutions based on multiple evolving parents naturally provide a self-adaptive feature to the algorithm [3]. Importantly, such an operation is missing in most other optimization algorithms and is one of the unique features of an EA. When it comes to solving real-parameter optimization problems, recombination between two real-valued vectors becomes more of a *blending* operation. Most existing real-parameter recombination operations can be classified according to whether the offspring solutions are created close to one of the parent solutions or close to the centroid of the participating parent solutions. We discuss these two classifications below.

2.1 Parent-Centric Crossover Operators

As the name suggests, in these recombination operators, the offsprings are created around the parents. The motivation behind these operators is that since parent solutions have been tested to be better in the preceding selection operation, creating offspring solutions near to parents may lead to better solutions. We discuss a couple of such operators.

2.1.1 Fuzzy Recombination (FR) Operator

This is a variable-wise recombination operator and was proposed by Voigt et al. in 1995 [9]. For two parent values of a particular variable, solutions close to each parent is chosen with a triangular probability distribution, as shown in

Figure 1. A parameter d is introduced to control

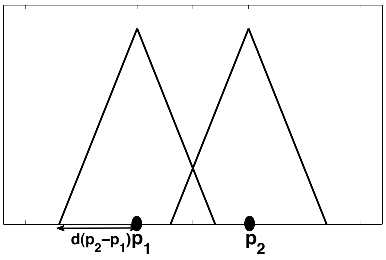


Figure 1: Fuzzy recombination (FR-d) operator. d is a user-defined parameter.

the diversity of the created offspring values vis-a-vis parent values.

2.1.2 Simulated Binary Crossover (SBX)

SBX is also a variable-wise recombination operator and was proposed by Deb et al. in 1995 [1]. This operator is similar to FR operator, but has the ergodic property such that any real value in the search space can be created from any two parent values, but with differing probabilities, as shown in Figure 2. The parameter η controls the diversity in the offspring values compared to that in the parent values.

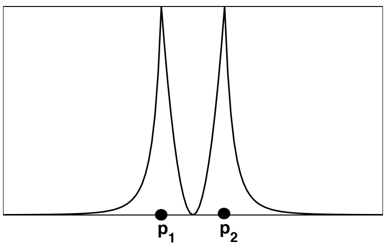


Figure 2: SBX-η operator. η is a user-defined parameter. Here we choose $\eta = 2$.

Since we have modified this operator in our study here, we describe this operator in somewhat more details. For two participating parent values (p_1 and p_2) of the i-th variable, two offspring values (c_1 and c_2) are created as a linear combination of parent values, as follows:

$$c_1 = 0.5(1-\beta(u))p_1 + 0.5(1+\beta(u))p_2, \quad (1)$$
$$c_2 = 0.5(1+\beta(u))p_1 + 0.5(1-\beta(u))p_2. \quad (2)$$

The parameter $\beta(u)$ depends on a random number u created within [0,1] for each recombination operation, and is given below:

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

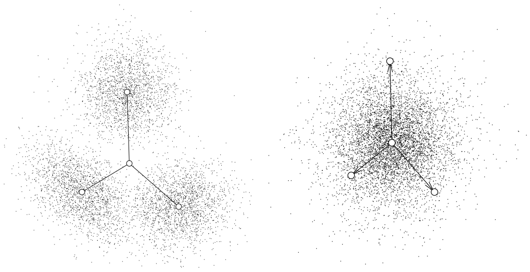
The resulting weights to p_1 and p_2 are biased in such a way that values close to the parent values are more likely than values away from them, thereby providing SBX its parent-centric property.

2.1.3 PCX Operator

The above variable-wise parent-centric operator can also be extended to vectors. One such operator is the parent-centric crossover or PCX [2], in which three parent solutions (vectors) are chosen 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:

$$\vec{y} = \mathbf{x}^{(p)} + w_\zeta \|\mathbf{d}^{(p)}\| \vec{e}^{(p)} + \sum_{i=1,\ i\neq p}^{n} w_\eta \bar{D} \vec{e}^{(i)}, \quad (4)$$

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 3(a) by creating a number of offspring solutions from three parent solutions.



(a) PCX operator. (b) UNDX operator.

Figure 3: PCX is a parent-centric and UNDX is a mean-centric operator.

2.2 Mean-Centric Crossover Operators

In these operators, offsprings 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 considered good. Some such operators are described below.

2.2.1 Blend Crossover (BLX)

BLX-α is a variable-wise recombination operator proposed by Eshelman and Schaffer [5]. Offsprings are created uniformly around the two parent values. The user-defined parameter α allows offsprings to be created within or outside the range of parents.

2.2.2 Vector-wise Mean-Centric Recombination Operators

In the unimodal normally distributed crossover (UNDX) operator (Figure 3(b)), three or more parents are chosen and points around the centroid of the parents are created with a multinomial normal distribution. Simplex crossover (SPX) [7] is another example of a mean-centric recombination operator.

3 Parent or Mean-Centric Recombination

Although both types of recombination operators exist, it is important to understand under what scenarios each type of recombination operator will be useful. Let us investigate two scenarios for this purpose.

Figure 4 depicts a scenario in which an EA population is approaching the optimum. In such a scenario, the population-best solution is likely to lie at the boundary of the current population, particularly when the population lies within the optimal basin. Ideally, in such a scenario, a parent-centric recombination is likely to yield better offspring solutions, as solutions beyond the population-best solution are expected to be closer to the optimum. Thus, while approaching an optimum, it is better to employ a parent-centric recombination operator for a faster approach towards the optimum.

On the other hand, Figure 5 shows a scenario in which the EA population surrounds the optimum and the population best is likely to be an intermediate population member. In such a case, a mean-centric recombination operator is likely to produce better solutions. Thus, while surrounding an optimum, it is better to employ a mean-centric recombination operator.

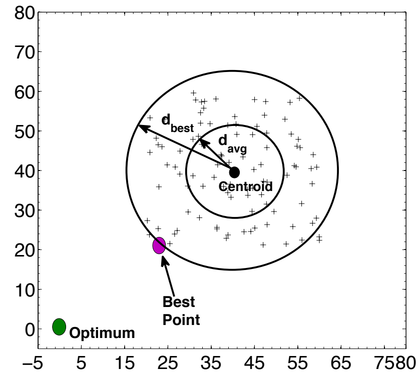


Figure 4: Scenario 1: Population is approaching the optimum. Here $\lambda > 1$.

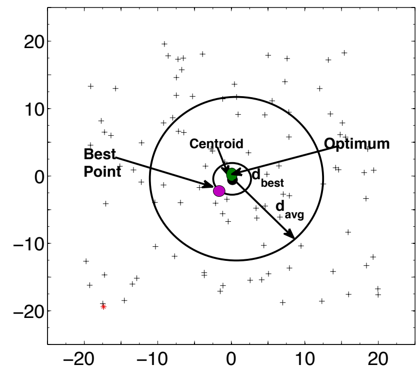


Figure 5: Scenario 2: Population surrounds the optimum. Here, $\lambda < 1$.

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 any particular generation so that a suitable type of recombination operator can be applied. In this study, we compute a parameter λ for this purpose based on the location of population-best and population-average points at every generation:

$$\lambda = \frac{d_{\text{best}}}{d_{\text{avg}}}, \tag{5}$$

where d_{best} is distance of the population-best solution from the centroid of population in the variable space and d_{avg} is the average distance of population members from the centroid of population. A little thought from Figures 4 and 5 will reveal that in the first scenario, λ is likely to be greater than one and in the second scenario it is likely to be smaller than one. Since the parameter λ captures the extent of approach or crowding near the optimum in the current population, we use this parameter to create an offspring based on parent-centric or mean-centric approach.

4 Self-Adaptive SBX (SA-SBX) Operator

We consider the SBX operator [1] and modify it using the λ value. First, we create two virtual parents v_1 and v_2 from two participating parents p_1 and p_2 as follows:

$$v_1 = \frac{p_1 + p_2}{2} - \lambda\frac{p_2 - p_1}{2}, \quad (6)$$

$$v_2 = \frac{p_1 + p_2}{2} + \lambda\frac{p_2 - p_1}{2}. \quad (7)$$

Next, the virtual parents (instead of p_1 and p_2) are then used to create two offspring values using equations 1 and 2. Unlike in another study [4] in which the SBX parameter η was updated from one generation to the other, here we keep it same from the start to the end of a simulation.

The above two equations yields:

$$(v_2 - v_1) = \lambda(p_2 - p_1).$$

Thus, if $\lambda = 0$ (population-best solution is at the centroid of the population), two virtual parents are identical and lie at the mean value of the two parent values. This causes a mean-centric operation. On the other hand, if $\lambda = 1$ (population-best solution is at the average distance from centroid of the population), virtual parents are identical to the original parents. This makes the SA-SBX to be identical to the original SBX operator. For $\lambda > 1$, virtual parents move outside the original parent values and the created offspring values are likely to have a greater diversity than that in the original SBX operation. Figure 6 illustrates three different scenarios.

Since the parameter λ is not pre-fixed and depends on the location of population-best solution from the centroid of the population and a measure of an average diversity of the population, λ value will change from one generation to another. If the population-best solution is at the periphery of the population (as in scenario 1 in Figure 4), SA-SBX creates a diverse set of offspring solutions and the search tends to explore the search space beyond the region represented by the population. Figure 7 shows the distribution of offspring values for a particular variable. However, when the population-best solution is close to the centroid of the population at any generation (as in scenario 2 in Figure 5), the above SA-SBX behaves like a mean-centric operator and diversity of the created offsprings is expected to reduce, thereby focusing the search. Figure 8 shows the corresponding distribution of offsprings. Interestingly, all these happen in a self-adaptive manner and without any user intervention. We now present simulation results using the SA-SBX operator.

5 Results

We choose five unimodal and three multi-modal test problems from the EA literature. They are tabulated in Table 1. To compare the performance of the proposed SA-SBX, we use the well-known CMA-ES [6] and 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 population of size $5n$ (where n is the number of variables in the problem), a crossover probability of 0.9, and $\eta = 2$ are used. To evaluate the effect of recombination operator alone, we have not used any mutation operator in this study. For all test problems, we have chosen two initialization processes: (i) symmetric initialization around optimum: $x_i \in [-20, 20]$ for all variables,

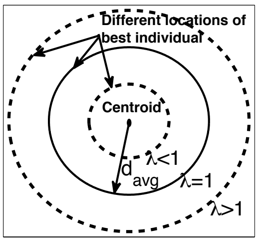


Figure 6: Parameter λ for different scenarios.

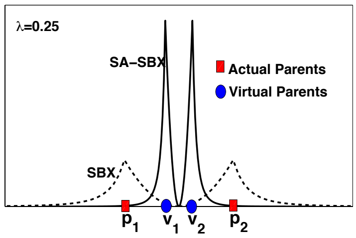


Figure 7: Distribution of offspring values for Scenario 1.

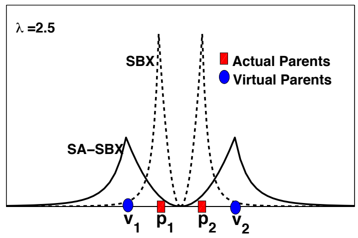


Figure 8: Distribution of offspring values for Scenario 2.

and (ii) one-sided initialization away from optimum: $x_i \in [20, 60]$ for all variables. Each problem is considered with different sizes, $n = 20$, 50, and 100.

5.1 Problem P1

Figure 9 shows the change in objective function value of the population-best member with generation for four different algorithms applied to the 20-variable version of P1. In this case, popula-

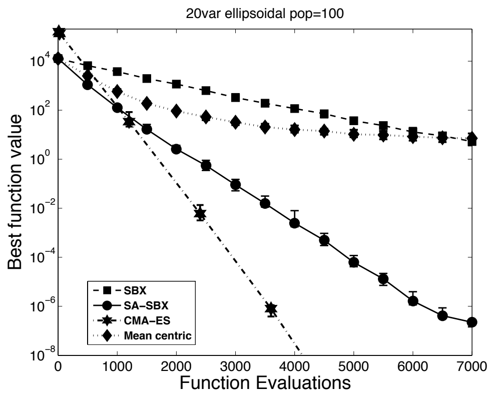


Figure 9: Reduction in objective value for 20-variable problem P1 with symmetric initialization.

tions are initialized around the optimum. It is clear that CMA-ES has outperformed all RGAs with any of the SBX variations. However, the enhancement in performance of SA-SBX compared to the original SBX is clear from the figure. The mean-centric version of the SBX operator did not perform so well. Interestingly, in all cases, the algorithms do not seem to be sensitive on 20 runs started from different initial populations.

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 10. Since

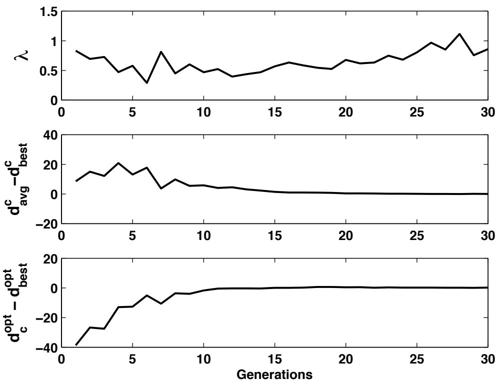


Figure 10: Variation of λ for 20-variable problem P1 with symmetric initialization.

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 how 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 differ-

Table 1: Unimodal and Multi-modal test problems used in the study.

Prob.	Objective function	f^*
	Unimodal Problems	
P1	$f_{\text{ellipsoidal}} = \sum_{i=1}^{n} i x_i^2$	0
P2	$f_{\text{discus}} = 10^4 x_1^2 + \sum_{i=2}^{n} x_i^2$	0
P3	$f_{\text{cigar}} = x_1^2 + 10^4 \sum_{i=2}^{n} x_i^2$	0
P4	$f_{\text{ridge}} = x_1^2 + 100\sqrt{\sum_{i=2}^{n} x_i^2}$	0
P5	$f_{\text{stepellip}} = 0.1 \max\left(\lvert s_1\rvert/10^4, \sum_{i=1}^{n} 10^{2\frac{i-1}{n-1}} s_i^2\right)$ $s_i = \begin{cases} \lfloor 0.5 + x_i \rfloor, & \text{if } x_i > 0.5, \\ \lfloor 0.5 + 10x_i \rfloor/10, & \text{otherwise.} \end{cases}$	0
	Multi-modal Problems	
P6	$f_{\text{rastrigin}} = \sum_{i=1}^{n} \left[x_i^2 + 10(1 - \cos(2\pi x_i))\right]$	0
P7	$f_{\text{griewangk}} = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} + 1\right)$	0
P8	$f_{\text{schaffer}} = \left(\frac{1}{n-1} \sum_{i=1}^{n-1} \sqrt{s_i} + \sqrt{s_i} \sin^2(50 s_i^{1/5})\right)^2$ where $s_i = \sqrt{x_i^2 + x_{i+1}^2}$	0

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

We now study the effect of one-sided initialization, in which the optimum is not surrounded by the initial population members. Figure 11 shows the performance of all four algorithms. Again, CMA-ES performs the best. The proposed SA-SBX performs better than the original SBX by a few orders of magnitude. Interestingly, the mean-centric version of SBX does not do as well as the original SBX, for the reason given earlier.

Since P1 is a unimodal problem, in early generations, one-sided initialization makes one

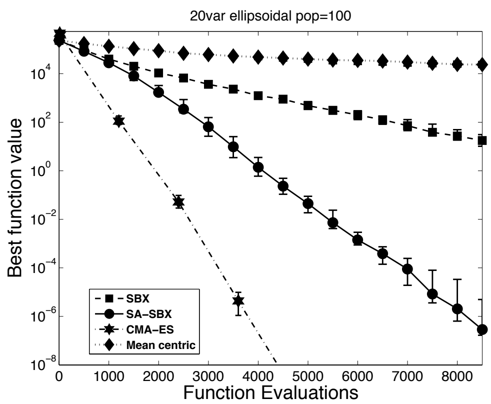


Figure 11: Reduction in objective value for 20-variable problem P1 with one-sided initialization.

of the boundary points as the population-best point. Hence, λ is expected to be greater than one early on. Figure 12 verifies this fact. Interestingly, this trend is opposite to that observed in the symmetric initialization case (Figure 10). 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 finds it to be better to reduce λ to enable convergence to the optimum.

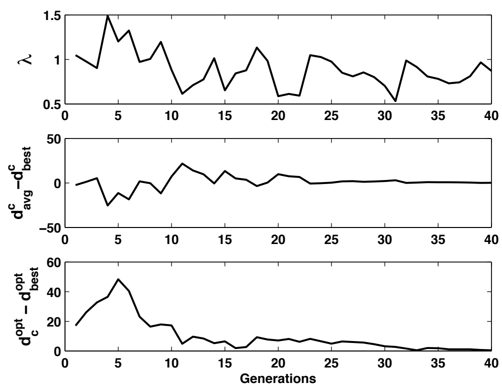


Figure 12: Variation of λ for 20-variable problem P1 with one-sided initialization.

50-variable and 100-variable versions of P1 are also attempted using all four methodologies using symmetric initializations. Table 2 tabulate the results. For the 50-variable problem with 16,471 function evaluations, the median objective value of 20 runs is found to be around $5(10^{-7})$, whereas using CMA-ES a median objective value of $3(10^{-14})$ is achieved with an identical number of function evaluations. Clearly, CMA-ES has performed better in this problem. Similar results are obtained for 100-variable version of the problem and also for one-sided initialization. For brevity, they are not discussed here.

5.2 Problems P2 to P5

Similar experiments are executed on the remaining unimodal test problems. The variation of population-best objective function value for symmetric and one-sided initializations with generation are plotted in Figures 13 and 16, respectively. In both cases, RGA with the proposed SA-SBX outperforms other algorithms. Results on 20 to 100-variable problems are tabulated in Table 2. Here, CMA-ES does not perform well particularly for 50 and 100-variable problems, for both symmetric and one-sided initializations, however the RGA with proposed SA-SBX performs better than other three algorithms.

CMA-ES performs well on P3 (cigar problem), however the proposed RGA with SA-SBX outperformed other three methods on P4 and P5, as they are clear from the table. In general, the original SBX fairs better than the mean-centric version of the SBX operator. Based on these extensive results, it can concluded that the RGA with the proposed SA-SBX performs consistently well on all unimodal test problems used in this study.

5.3 Multi-modal Problems P6 to P8

Figures 14 and 17 show the reduction in objective function value with generations for symmetric and one-sided initializations on 100-variable version of P6. The Rastrigin function (P6) has many local optimum and the proposed SA-SBX operator through its parent-centric to mean-centric adaptations helps to avoid all local optimum even when initialized around $x_i \in [20, 60]$ (far away from global basin) and converge to the global optimum. The variation of λ in a typical run for the one-sided initialization is shown in Figure 19. In all 20 runs, the performance is

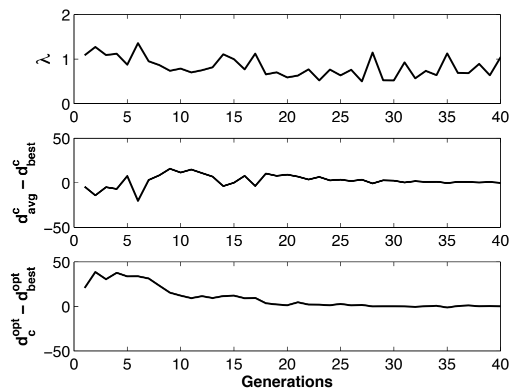


Figure 19: Variation of λ for 20-variable problem P6 with one-sided initialization.

quite similar, thereby demonstrating the robustness of the proposed approach. Based on the current location of population-best and population diversity, a focused or a broader search takes place. While converging to a local optimum, as soon as a better solution on a better local optimal basin is found, a large value of λ is obtained and a broader search takes place, thereby taking the population out of the current local basin. The fluctuating λ value explains this phenomenon. All other algorithms including CMA-ES seem to get stuck to a local optimum. RGAs with the original SBX operator could never find

Table 2: Objective function value (median of 20 runs) obtained after a fixed allocated number of function evaluations for all four algorithms are shown.

Function	Initialization	n	Func. evals.	SA-SBX	parent-centric SBX	mean-centric SBX	CMA-ES
$f_{\text{ellipsoidal}}$	$[-20, 20]^n$	20	5,893	1.65×10^{-6}	1.29×10^{1}	8.39	$\mathbf{3.48 \times 10^{-15}}$
		50	16,471	5.09×10^{-7}	1.06×10^{3}	8.41×10^{1}	$\mathbf{2.69 \times 10^{-14}}$
		100	40,000	1.56×10^{-7}	1.25×10^{4}	5.39×10^{2}	$\mathbf{1.17 \times 10^{-14}}$
	$[20, 60]^n$	20	5,989	1.42×10^{-3}	1.20×10^{2}	3.29×10^{3}	$\mathbf{1.35 \times 10^{-14}}$
		50	17,521	8.69×10^{-5}	4.62×10^{3}	7.78×10^{4}	$\mathbf{3.45 \times 10^{-15}}$
		100	39,781	1.34×10^{-4}	5.96×10^{4}	1.76×10^{5}	$\mathbf{1.15 \times 10^{-14}}$
f_{discus}	$[-20, 20]^n$	20	8,000	$\mathbf{1.00 \times 10^{-9}}$	1.16×10^{-7}	2.15	5.34×10^{-4}
		50	17,500	$\mathbf{1.61 \times 10^{-7}}$	8.99×10^{1}	9.44	3.00×10^{2}
		100	37,500	$\mathbf{1.48 \times 10^{-7}}$	5.16×10^{2}	2.16×10^{1}	7.53×10^{2}
	$[20, 60]^n$	20	10,000	1.00×10^{-8}	2.78	1.09×10^{4}	$\mathbf{8.56 \times 10^{-9}}$
		50	22,500	$\mathbf{1.74 \times 10^{-7}}$	2.08×10^{2}	1.21×10^{4}	1.05×10^{1}
		100	47,500	$\mathbf{1.54 \times 10^{-7}}$	1.60×10^{3}	1.29×10^{4}	1.22×10^{2}
f_{cigar}	$[-20, 20]^n$	20	6,805	1.82×10^{-4}	4.26	3.81×10^{3}	$\mathbf{8.71 \times 10^{-9}}$
		50	15,406	1.77×10^{-3}	6.08×10^{5}	4.54×10^{4}	$\mathbf{8.95 \times 10^{-9}}$
		100	30,261	5.19×10^{-3}	5.01×10^{6}	1.63×10^{5}	$\mathbf{9.46 \times 10^{-9}}$
	$[20, 60]^n$	20	6,685	2.736×10^{-1}	5.36×10^{4}	2.48×10^{7}	$\mathbf{8.65 \times 10^{-9}}$
		50	16,171	1.44	3.63×10^{6}	4.79×10^{7}	$\mathbf{9.49 \times 10^{-9}}$
		100	30,227	4.33	2.38×10^{7}	6.95×10^{7}	$\mathbf{9.70 \times 10^{-9}}$
$f_{\text{sharpridge}}$	$[-20, 20]^n$	20	19,500	$\mathbf{5.94 \times 10^{-7}}$	2.22	1.85×10^{1}	1.32
		50	33,750	$\mathbf{5.85 \times 10^{-7}}$	1.11×10^{2}	1.35×10^{2}	8.99×10^{-1}
		100	70,000	$\mathbf{6.74 \times 10^{-7}}$	6.09×10^{1}	2.55×10^{2}	2.45×10^{-1}
	$[20, 60]^n$	20	22,500	$\mathbf{5.17 \times 10^{-7}}$	2.43	1.74×10^{3}	4.85×10^{-1}
		50	37,500	$\mathbf{5.68 \times 10^{-7}}$	2.04×10^{2}	2.61×10^{3}	5.31×10^{-1}
		100	80,000	$\mathbf{6.22 \times 10^{-7}}$	1.02×10^{3}	2.19×10^{3}	3.93×10^{-1}
$f_{\text{stepellisoidal}}$	$[-20, 20]^n$	20	3,500	$\mathbf{2.00 \times 10^{-8}}$	2.00×10^{-3}	1.28	1.63×10^{-1}
		50	8,750	$\mathbf{4.48 \times 10^{-7}}$	6.80×10^{-2}	6.64	1.83
		100	20,000	$\mathbf{1.00 \times 10^{-9}}$	1.09×10^{-1}	1.60×10^{11}	5.03
	$[20, 60]^n$	20	5,500	$\mathbf{2.00 \times 10^{-8}}$	1.77×10^{1}	2.10×10^{3}	4.02×10^{-1}
		50	13,750	$\mathbf{3.70 \times 10^{-8}}$	2.71×10^{2}	3.07×10^{3}	1.45
		100	30,000	$\mathbf{1.00 \times 10^{-9}}$	1.29×10^{3}	4.19×10^{3}	8.98
$f_{\text{rastrigin}}$	$[-20, 20]^n$	20	11,250	$\mathbf{1.54 \times 10^{-7}}$	5.72×10^{1}	1.93×10^{1}	2.59×10^{1}
		50	20,000	$\mathbf{2.02 \times 10^{-7}}$	3.98×10^{2}	9.97×10^{1}	1.04×10^{2}
		100	42,500	$\mathbf{1.44 \times 10^{-7}}$	1.13×10^{3}	3.26×10^{2}	2.67×10^{2}
	$[20, 60]^n$	20	5,665	$\mathbf{1.12 \times 10^{-7}}$	6.02×10^{1}	8.01×10^{2}	2.88×10^{1}
		50	26,250	$\mathbf{1.41 \times 10^{-7}}$	4.28×10^{2}	2.26×10^{3}	9.95×10^{1}
		100	52,500	$\mathbf{1.61 \times 10^{-7}}$	1.49×10^{2}	2.57×10^{3}	2.56×10^{2}
f_{schaffer}	$[-20, 20]^n$	20	12,000	$\mathbf{3.43 \times 10^{-7}}$	2.49×10^{-2}	4.23×10^{-2}	8.02×10^{-2}
		50	21,250	$\mathbf{4.20 \times 10^{-7}}$	2.82×10^{-1}	1.86×10^{-1}	4.22×10^{-1}
		100	45,000	$\mathbf{3.77 \times 10^{-7}}$	8.83×10^{-1}	2.97×10^{-1}	1.24
	$[20, 60]^n$	20	14,250	$\mathbf{3.94 \times 10^{-7}}$	3.04×10^{-2}	3.99	9.65×10^{-2}
		50	26,250	$\mathbf{3.66 \times 10^{-7}}$	3.93×10^{-1}	4.66	1.13
		100	52,500	$\mathbf{3.59 \times 10^{-7}}$	1.33	3.48	3.04
$f_{\text{griewangk}}$	$[-20, 20]^n$	20	5,509	1.07×10^{-5}	6.60×10^{-1}	5.26×10^{-2}	$\mathbf{4.44 \times 10^{-15}}$
		50	14,146	1.13×10^{-7}	9.96×10^{-1}	1.65×10^{-1}	$\mathbf{6.00 \times 10^{-15}}$
		100	30,000	1.38×10^{-7}	1.17	2.46×10^{-1}	$\mathbf{3.06 \times 10^{-14}}$
	$[20, 60]^n$	20	5,665	1.04×10^{-2}	1.03	1.88	$\mathbf{2.66 \times 10^{-15}}$
		50	14,596	2.79×10^{-5}	1.13	2.32	$\mathbf{9.10 \times 10^{-15}}$
		100	30,465	9.30×10^{-6}	1.59	2.66	$\mathbf{3.70 \times 10^{-14}}$

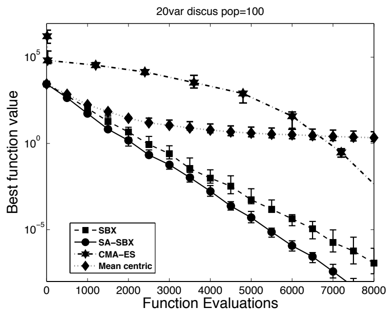


Figure 13: Population-best objective value for 20-variable problem P2 with symmetric initialization.

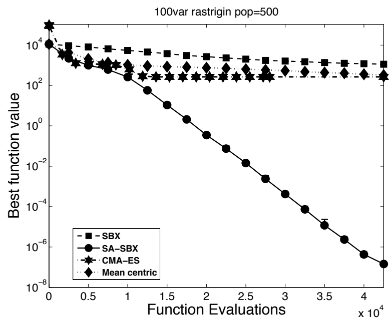


Figure 14: Population-best objective value for 100-variable problem P6 with symmetric initialization.

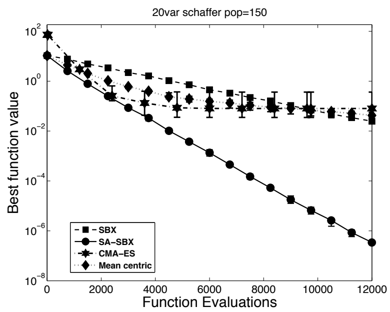


Figure 15: Population-best objective value for 20-variable problem P8 with symmetric initialization.

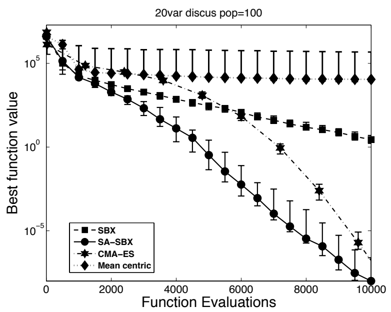


Figure 16: Population-best objective value for 20-variable problem P2 with one-sided initialization.

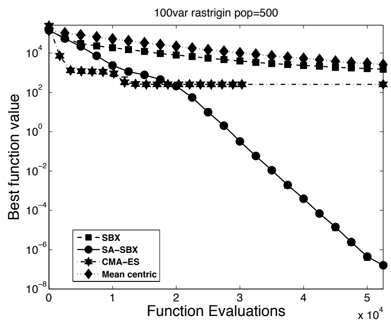


Figure 17: Population-best objective value for 100-variable problem P6 with one-sided initialization.

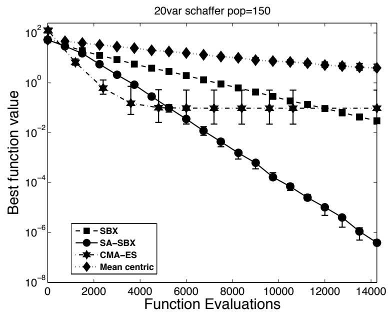


Figure 18: Population-best objective value for 20-variable problem P8 with one-sided initialization.

the global optimum for this problem so far. Similar results of better performance of the proposed approach are obtained for problem P8 as well (Figures 15 and 18).

Table 2 tabulates the median objective values obtained in 20 runs for both symmetric and one-sided initializations for all three multi-modal problems. In all three problems, RGA with adaptive SBX performs consistently well, whereas CMA-ES performs better only in problem P7 (Griewangk).

6 Conclusions

In this paper, we have argued and demonstrated that instead of using a purely parent-centric or a purely mean-centric recombination operator, an adaptive use of them in a real-parameter optimization algorithm is a better overall approach. By modifying the existing simulated binary crossover (SBX) operator with a self-adaptive parameter that depends on the location of population-best solution compared to the diversity associated with the population, a self-adaptive transition from parent-centric to mean-centric versions of the SBX recombination operator has been achieved. On a number of uni-modal and multi-modal test problems having 20, 50, and 100 variables, the real-parameter genetic algorithm (RGA) with SA-SBX has consistently and reliably found the global minimum in most problems, whereas other algorithms including the well-known CMA-ES could not solve all problems for the allocated number of function evaluations.

This proof-of-principle study shows promise for a more detailed study in which (i) a number of other more complex problems can be taken up, (ii) the principle of self-adaptive transition from parent-centric to mean-centric operations can be extended to multi-objective optimization problems, (iii) instead of basing the λ parameter on the population-best solution alone, the top 5% or 10% population members may be considered for this purpose. We are currently pursuing some of these extensions, Nevertheless, this study has shown a clear indication of the importance of an adaptive approach compared to either purely parent-centric or purely mean-centric recombination approaches.

References

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

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

[3] K. Deb and H.-G. Beyer. Self-adaptation in real-parameter genetic algorithms with simulated binary crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pages 172–179, 1999.

[4] K. Deb, K. Sindhya, and T. Okabe. Self-adaptive simulated binary crossover for real-parameter optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*, pages 1187–1194. New York: The Association of Computing Machinery (ACM), 2007.

[5] 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.

[6] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation Journal*, 9(2):159–195, 2000.

[7] T. Higuchi, S. Tsutsui, and M. Yamamura. Theoretical analysis of simplex crossover for real-coded genetic algorithms. In *Parallel Problem Solving from Nature (PPSN-VI)*, pages 365–374, 2000.

[8] 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.

[9] 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.