A Novel Crossover Operator Designed to Exploit Synergies of Two Crossover Operators for Real-Coded Genetic Algorithms

Shashi and Kusum Deep

Abstract In this paper a new crossover operator called the double distribution crossover (DDX) is proposed. The performance of DDX is compared with existing real-coded crossover operator namely Laplace crossover (LX). DDX is used in conjunction with a well-known mutation operator; Power mutation (PM) to obtain a new generational real-coded genetic algorithm called DDX-PM. DDX-PM is compared with the existing LX-PM. The performance of both the genetic algorithms is compared on the basis of success rate, average function evaluation, average error and computational time, and the preeminence of the proposed crossover operator is established.

Keywords Real-coded genetic algorithm · Mutation operator · Crossover operator

1 Introduction

The demand for designing efficient, reliable and robust numerical optimization techniques is very high because of their ability to tackle a number of real-life complex nonlinear optimization problems arising in the field of Engineering, Science, Industry and Finance, which may have constraints associated with them. There are a number of stochastic optimization techniques which are being used to solve complex optimization problems. Among these genetic algorithms (GA) are found to be very promising global optimizers. GAs are population-based heuristics which mimic the Darwin's principal of "survival of fittest". The concept of GA was given by Holland [1]. A detailed implementation of GA could be found in Goldberg

Shashi (✉)
Control and Decision Systems Laboratory, Department of Aerospace
Engineering, Indian Institute of Science, Bangalore, India
e-mail: shashibarak@gmail.com

K. Deep
Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, India
e-mail: kusumfma@iitr.ernet.in

M. Pant et al. (eds.), *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 437, DOI 10.1007/978-981-10-0451-3_32

[2]. The main attribute of GAs is their easy implementation nature, as they do not require any extra information such as continuity and differentiability about objective function and constraints. Although initial versions of GA use binary numbers for encoding of chromosomes, their major drawbacks such as lack of precision and existence of “Hamming-cliff” problem impel to look for another type of encoding scheme. To overcome these difficulties related to binary encoding of continuous parameter optimization problems, real encoding of chromosomes is used. GAs which make use of real encoding of chromosomes are termed as real-coded GA (RCGA).

In GAs crossover is considered to be main search operator. The crossover operator is used to thoroughly explore the search process. After the selection process, the population is enriched with better individuals. Selection makes clones of good strings but does not create new ones. Crossover operator is applied to the mating pool with the hope that it creates better offsprings. In crossover operator the genetic information between two or more individuals is blended to produce new individuals.

Though, along with crossover mutation operator is also essential for thorough search of the function landscape. The most important aspect of mutation operator is that it prevents premature convergence of an algorithm. Mutation operator provides random diversity in the population as described in [3].

A lot of effort has been put into the development of sophisticated real-coded crossover operators [4–8] to improve the performances of RCGAs for function optimization. Deep and Thakur [9] presented Laplace crossover operator which generates a pair of offspring solution from a pair of parent solutions using Laplace distribution. Tutkun [10] proposed a crossover operator based on Gaussian distribution. Kaelo and Ali [11] suggested integration of different crossover rules in the genetic algorithm and recommended some modifications in applying the crossover rules and localization of searches in their study. Recent development in RCGA includes [12].

In an endeavour to define new operators for real-coded genetic algorithms, in this paper, a new crossover operator called double distribution crossover (DDX) is presented. Combining DDX with power mutation (PM) [13], a new generational RCGA called DDX-PM is designed. It is compared with existing RCGA, LX-PM. In order to establish the strength of double distribution crossover computational analysis is performed and discussed.

The paper is organized as follows: the proposed double distribution crossover is defined in Sect. 2. In Sect. 3, new RCGAs based on double distribution crossover are discussed. In Sect. 4, the experimental setup of RCGAs is explained. The computational results and discussion is given in Sect. 5. Finally, in Sect. 6, the conclusions are drawn.

2 The Proposed Crossover Operator

In this section a new crossover operator based on the novel idea of merging of two existing crossover operators is proposed in such a way that it retains the strength of the operators from which it is obtained. The proposed crossover operator is called as double distribution crossover (DDX) as it makes use of two well-established crossover operators: Laplace Crossover (LX) [9] based on Laplace distribution and Weibull Crossover (WX) [14] based on Weibull distribution to produce two offsprings and hence named as double distribution crossover (DDX) operator.

Although DDX could be obtained by merging any two crossover operators, here WX and LX are used because they are well-established potentially effective crossover operators. In DDX two offsprings O_1 and O_2 are generated from a pair of parents P_1 and P_2 obtained after selection, in the following manner:

Generate a uniformly distributed random number $u \in [0, 1]$
if $u \leq \frac{1}{2}$; then

$$\begin{aligned} O_1 &= P_1 + \mu d \\ O_2 &= P_2 + \mu d \end{aligned} \tag{1}$$

else

$$\begin{aligned} O_1 &= P_1 - \mu d \\ O_2 &= P_2 - \mu d \end{aligned} \tag{2}$$

where $d = |P_1 - P_2|$ is the distance between the parents and μ is calculated as $\mu = \frac{r_1 + r_2}{2}$, where r_1 is a random number following Laplace distribution and r_2 is a random number following Weibull distribution. The density function of Weibull distribution is given as follows:

$$f(x) = \begin{cases} ab^{-a}x^{a-1}e^{-(x/b)^a} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and the distribution function is given by

$$F(x) = \begin{cases} 1 - e^{-(x/b)^a} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $a > 0$ is called shape parameter and $b > 0$ is called the scale parameter. And the density function of Laplace distribution is given as follows:

$$f(x) = \frac{1}{2d}\exp\left(-\frac{|x-c|}{d}\right), \quad -\infty < x < \infty \tag{5}$$

the distribution function is given by

$$F(x) = \begin{cases} \frac{1}{2}\exp\left(\frac{|x-c|}{d}\right) & \text{if } x \leq c \\ 1 - \frac{1}{2}\exp\left(-\frac{|x-c|}{d}\right) & \text{if } x > c \end{cases} \quad (6)$$

where $c \in R$ is location parameter and $d > 0$ is termed as scale parameter.

DDX preserves the property of both the parent operators, i.e. for fixed values of the crossover index of DDX (which is a two-dimensional vector which includes the crossover indices of both WX and LX), the spread of offsprings is proportional to the spread of parents, i.e. if the parents are near to each other the offsprings are expected to be near to each other and if the parents are far from each other then offsprings are likely to be far from each other.

Also, from Eqs. 1 and 2 it is clear that both the offsprings are placed symmetrical with respect to the position of the parents and if DDX produces an offspring which is not within the bounds of the decision variable then that offspring is assigned a random value within its bounds.

3 The Proposed New RCGA

In this section a new RCGA namely DDX-PM is proposed, which combines the use of Laplace crossover with power mutation [13]. DDX-PM is compared with the existing LX-PM. Both the algorithms use tournament selection with tournament size three. In addition, both the RCGAs are elite preserving with elitism size one. The elitism operator helps to maintain fitness stability to increase the search performance of the proposed algorithm.

Both the RCGAs are terminated if either the pre-specified maximum number of generations (3000) is reached or the best solution obtained by the algorithm lies within specified accuracy (0.01) of known optimum, whichever occurs earlier. The algorithm is tested on standard benchmark problems taken from literature which includes five nonscalable test problems (Table 1) and five scalable test problems of

Table 1 List of nonscalable benchmark functions

S. no.	Function	Mathematical formula	Range	Min
1	Easom 2D	$\text{Min} f(x) = -\cos(x_1)\cos(x_2)\exp\left(-(x_1-\pi)^2-(x_2-\pi)^2\right)$	[−10, 10]	−1
2	Becker and lago	$\text{Min} f(x) = (\lvert x_1\rvert - 5)^2 + (\lvert x_2\rvert - 5)^2$	[−10, 10]	0
3	Bohachevsky 1	$\text{Min} f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	[−50, 50]	0
4	Eggcrate	$\text{Min } f(x) = x_1^2 + x_2^2 + 25(\sin^2 x_1 + \sin^2 x_2)$	[−2 π, 2 π]	0
5	Periodic	$\text{Min} f(x) = 1 + \sin^2 x_1 + \sin^2 x_2 - 0.1\exp(-x_1^2 - x_2^2)$	[−10, 10]	0.9

size 30 (Table 2) of different levels of complexity and multimodality. The pseudocode of DDX-PM is given below:

Algorithm

```
    Step 1 (Initialization):
      • Initialize population;
      • Set Generation=0;
Step 2(Evaluation): Evaluate the fitness for each individual
Step 3(Termination check): Check the termination criteria
        If satisfied stop;
          else goto 4.
Step 4 (GA Operations)
      • Select individuals according to selection algorithm to build a mating pool
      • Crossover the population in mating pool with given crossover probability
      • Mutate the current population with given mutation probability
Step 5 (Replacement): Replace the old population with new population while retaining the
best individual for next generation i.e. apply elitism with size 1
Step 6
      • Evaluate the best fitness and find optimal individual
      • Increment generation; go to step 3.
```

4 Experimental Setup

The experimental setup used for both the RCGAs viz. DDX-PM and LX-PM is given in this section.

Parameter Setting: Population size is taken as 10 times the number of variables. The crossover index for DDX is fixed at (12, 0.20). The final parameter setting is presented in Table 3 where p_c and p_m represent the probabilities of crossover and mutation, respectively. Each GA runs at 100 times with same initial populations while each run is initiated using a different set of initial population.

All the algorithms are implemented in C++ and the experiments are done on a Core Duo Processor with 1.66 GHz speed and 1 GB RAM under WINXP platform.

Performance Evaluation Criteria: A run in which the algorithm finds a solution satisfying $f_{\min} - f_{\text{opt}} \leq 0.01$, where $f_{\min}$ is the best solution found when the algorithm terminates and f_{opt} is the known global minimum of the problem, is considered to be successful.

For each method and problem, the followings are recorded:

- Success Rate $(\text{SR}) = \frac{\text{Number of successful runs}}{\text{Total number of runs}} \times 100$
- Average computational time (ACT) (in seconds).
- Average number of function evaluations (AFE).
- Average Error $(\text{AE}) = \frac{\sum_n \left(f_{\min} - f_{\text{opt}}\right)}{n}$ where, n is the total number of runs.

Table 2 List of scalable benchmark functions

S. no.	Function	Mathematical formula	Range	Min
1	Paviani	$\min_x f(x) = \sum_{i=1}^{n} \left[(\ln(x_i - 2))^2 + (\ln(10 - x_i))^2\right] - \left(\prod_{i=1}^{n} x_i\right)^{0.2}$	$[2, 10]^{30}$	−997807.705158
2	Rastrigin	$\min_x f(x) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	$[-5.12, 5.12]^{30}$	0
3	Schwefel	$\min_x f(x) = -\sum_{i=1}^{n} x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	$[-500, 500]^{30}$	−12569.48
4	Sinusoidal	$\min_x f(x) = -\left[2.5 \prod_{i=1}^{n} \sin\left(x_i - \frac{\pi}{6}\right) + \prod_{i=1}^{n} \sin\left(5\left(x_i - \frac{\pi}{6}\right)\right)\right]$	$[0, \pi]^{30}$	−3.5
5	Zakharov's	$\min_x f(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} \frac{i}{2} x_i\right)^2 + \left(\sum_{i=1}^{n} \frac{i}{2} x_i\right)^4$	$[-10, 10]^{30}$	0

Table 3 Parameter setting for DDX-PM and LX-PM

Algorithm	Nonscalable		Scalable	
	p_c	p_m	p_c	p_m
DDX-PM	0.70	0.040	0.60	0.010
LX-PM	0.70	0.007	0.60	0.005

Table 4 Computational results for nonscalable and scalable problems

Nonscalable problems					Scalable problems				
Fun	Method	AE	SR	AFE	Fun	AE	SR	AFE	ACT
1	LX-PM	0.00625	100	378	1	0.00865	100	139,364	3.43
	DDX-PM	0.00478	100	332		0.00214	100	57,300	0.98
2	LX-PM	0.00612	100	4059	2	0.01154	77	883,517	31.35
	DDX-PM	0.00583	100	339		0.00634	100	59,476	1.78
3	LX-PM	0.00884	94	649	3	0.00756	100	128,034	2.82
	DDX-PM	0.0048	100	597		0.00632	88	125,463	3.25
4	LX-PM	0.01102	66	12,718	4	0.00805	100	46,486	1.06
	DDX-PM	0.00478	100	387		0.00953	100	17,116	0.42
5	LX-PM	0.02115	44	1394	5	0.00874	100	336,236	6.82
	DDX-PM	0.00598	82	926		0.00966	100	86,225	2.06

5 Results and Discussion

The computational results of both the RCGAs are presented in this section.

Table 4 summarizes the computational results for both the algorithms. As the execution time for nonscalable problems is very less and hence insignificant in most of the cases, it is not included in Table 4. From Table 4 it can be easily observed that the proposed crossover operator DDX outperforms LX in terms of various performance criteria and thus indicates the efficiency of proposed crossover operator for providing suitable exploration for obtaining the global optimal solution.

6 Conclusion

In this paper a new crossover operator called double distribution crossover (DDX) is introduced. By combining DDX with already existing operator, namely PM a new generational RCGA called DDX-PM is proposed. For an analogical comparison it is compared with the existing RCGA based on Laplace crossover and power mutation called LX-PM. Based on the numerical results it is clear that with respect to *reliability*, *efficiency* and *accuracy* measured in terms of success rate, function evaluation and average error, respectively, DDX-PM performed better than LX-PM, though they both use same mutation operator, which clearly signifies the role of crossover operators in the search process for locating global optima.

Thus DDX proved to be an efficient crossover operator when used with power mutation, but whether its performance will remain equally well, when used with other mutation operators for RCGAs, is the question whose answer is to be looked in future studies.

Acknowledgments Shashi thankfully acknowledge the financial assistance from National Board of Higher Mathematics, Department of Atomic Energy, Government of India.

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan press, Ann Arbor (1975)
2. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, New York (1989)
3. Spears, W.M.: Crossover or mutation? In: Whitley, L.D. (ed.) Foundations of Genetic Algorithms 2, pp. 221–238. Morgan Kaufmann, San Mateo, CA (1993)
4. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Syst. **9**, 115–148 (1995)
5. Hong, I., Kahng, A.B., Moon, B.R.: Exploiting synergies of multiple crossovers: initial studies. In: Proceedings of Second IEEE International Conference on Evolutionary Computation, IEEE Press, Piscataway, NJ, pp. 245–250 (1995)
6. Yoon, H.S., Moon, B.R.: An empirical study on the synergy of multiple crossover operators. IEEE Trans. Evol. Comput. **6**(2), 212–223 (2002)
7. Herrera, F., Lozano, M., Sanchez, A.M.: Hybrid crossover operators for real coded genetic algorithms: an experimental study. Soft. Comput. **9**(4), 280–298 (2005)
8. Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter evolution. Evol. Comput. J. **10**(4), 371–395 (2002)
9. Deep, K., Thakur, M.: A new crossover operator for real coded genetic algorithms. Appl. Math. Comput. **188**, 895–911 (2007)
10. Tutkun, N.: Optimization of multimodal continuous functions using a new crossover for the real-coded genetic algorithms. Expert Syst. Appl. **36**, 8172–8177 (2009)
11. Kaelo, P., Ali, M.M.: Integrated crossover rules in real coded genetic algorithms. Eur. J. Oper. Res. **176**, 60–76 (2007)
12. Thakur, M.: A new genetic algorithm for global optimization of multimodal continuous functions. J. Comput. Sci. **5**, 298–311 (2014)
13. Deep, K., Thakur, M.: A new mutation operator for real coded genetic algorithms. Appl. Math. Comput. **193**, 229–247 (2007)
14. Deep, K., Shashi., Katiyar, V.K.: Global optimization of Lennard-Jones potential using newly developed real coded genetic algorithms. In: Proceedings of 2011—International Conference on Communication Systems and Network Technologies, (IEEE Computer Society Proceedings), pp. 614–618 (2011)