Research Journal of Applied Sciences 10 (3): 75-79, 2015
ISSN: 1815-932X

Combined Crossover Operator

[1]Khalid Jebari, [1]Amina Dik, [1,2]Abdelaziz Bouroumi and [1]Aziz Ettouhami
[1]LCS Laboratory, Faculty of Sciences,
Mohammed V-Agdal University, UM5A Rabat, Morocco
[2]LTI Information Treatment Laboratory, Ben M'sik Faculty of Sciences,
Hassan II Mohammedia University (UH2M), Casablanca, Morocco

Abstract: Genetic algorithms are optimization and search methods based on the principles of Darwinian evolution and genetics that try to provide the optimal solution of a problem. They evolve a population of candidate solutions to the problem, using mutation, crossover and selection operators. Based on the diversity and the efficiency of four well known crossover operators, this study presents a novel operator called Combined Crossover Operator (CCO). The comparison with those four crossover operators shows that the results obtained by the CCO are promising.

Key words: Evolutionary computation, crossover, genetic algorithms, real coding crossover, CCO

INTRODUCTION

Genetic Algorithms (GAs) represent search and optimization technique based on principles of Mendel and Darwin. In his book, "Adaptation in Natural and Artificial Systems" (Holland, 1975), Holland described how to apply the principles of natural evolution to optimization problems and built the first GAs. Sine, the emergence of Goldberg (1989) book's "Genetic Algorithms in Search, Optimization and Machine Learning", the GAs became increasingly a powerful tool for solving search and optimization problems. In Gas, a population of candidate solutions called chromosomes randomly chosen is evolved through generations according to mechanisms of selection, crossover and mutation. The two most commonly employed genetic search operators are crossover and mutation. Crossover produces off spring by recombining the information from two parents (Deep and Thakur, 2007). Mutation prevents convergence of the population by flipping a small number of randomly selected bits to continuously introduce variation. The driving force behind GAs is the unique cooperation between selection, crossover and mutation operator. A genetic operator is a process used in GAs to maintain genetic diversity (Poli and Langdon, 2006). The most widely used genetic operators are crossover and mutation.

The Combined Crossover Operator (CCO) is a uniform crossover operator which has a different maximum range of variation depending on the quality of the solution. The main idea of the operator is to use good parents (efficient ones) to improve the quality of the offspring (exploitation) and to use not so good parents (non-efficient ones) to explore the whole space (exploration).

Literature review: The crossover operator is a genetic operator that combines two chromosomes (parents) to produce a new chromosome (Child) (Lobo *et al*., 2007). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-definable crossover probability (Reeves and Rowe, 2002). For purposes of this work, only crossover operators that operate on two parents and have no self adaptation properties will be considered.

Heuristic crossover: In Heuristic Crossover (HC) (Ortiz-Boyer *et al*., 2005), heuristic returns an offspring that lies on the line containing the two parents, a small distance away from the parent with the best fitness value in the direction away from the parent with the worse fitness value. The default value of ratio is 1.2. If $Parent^1$ and $Parent^2$ are the parents and $Parent^1$ has the best fitness value, the function returns the child (Herrera and Lozano, 1996):

$$Child_i^1 = Parent_i^2 + Ratio \times \left(Parent_i^1 - Parent_i^2\right) \quad (1)$$

$$Child_i^2 = Parent_i^1 + Ratio \times \left(Parent_i^1 - Parent_i^2\right) \quad (2)$$

Corresponding Author: Khalid Jebari, LCS Laboratory, Faculty of Sciences, Mohammed V-Agdal University, UM5A Rabat, Morocco

Arithmetic crossover: In Arithmetic Crossover (AC) (Herrera *et al*., 2003), arithmetic creates children that are the weighted arithmetic mean of two parents. Children are feasible with respect to linear constraints and bounds. Alpha is a random value between [0, 1]. If $Parent^1$ and $Parent^2$ are the parents and $Parent^1$ has the best fitness value, the function returns the child (Hong *et al*., 2002; Zbigniew, 1996), according the equation:

$$Child_i^1 = \alpha\times Parent_i^1 + (1-\alpha)\times Parent_i^2 \quad (3)$$

$$Child_i^2 = \alpha\times Parent_i^2 + (1-\alpha)\times Parent_i^1 \quad (4)$$

Simulated binary crossover: This crossover operator works with two parent solutions and creates the offspring. SBX (Deb and Agrawal, 1995) simulates the working principle of the single-point crossover operator on binary strings. During this operation common interval schemata between the parents are preserved in the offspring. The procedure of computing the offspring $Child_i^1$ and $Child_i^2$ from the parents $Parent_i^1$ and $Parent_i^2$ is described as follows.

A spread factor β_i is obtained as the ratio of the absolute difference in the offspring values to their parents:

$$\beta_i = \left|\frac{Child_i^2 - Child_i^1}{Parent_i^2 - Parent_i^1}\right| \quad (5)$$

With the above definition of the spread factor, crossover operators are classified as:

- Contracting crossovers $\beta_i<1$: the parent points enclose the children points
- Expanding crossovers $\beta_i>1$: the children points enclose the parent points
- Stationary crossovers $\beta_i = 1$: the children points are the same as the parent points

First a random number $\mu\in[0, 1]$ is created. From a specified probability distribution function, the ordinate β_α is found so that the area under the probability curve from 0 to β_q is equal to the chosen random number μ. The probability distribution used to create a child's solution is derived to have a similar search power as that in a single-point crossover in binary coded GAs and is given as follows (Deb and Kumar, 1995):

$$P(\beta_i) = \begin{cases} 0.5(\eta+1)\beta^{\eta} & \text{if } \beta \le 1 \\ 0.5(\eta+1)\beta^{\frac{1}{\eta+2}} & \text{otherwise} \end{cases} \quad (6)$$

In the above expressions, the distribution index η gives a higher probability for creating near parent solutions as a small value of η allows distant solutions to be selected as children solutions. Using Eq. 6, β_q is calculated by equating the area under the probability curve to μ as follows:

$$\beta_q = \begin{cases} (2\mu)^{\frac{1}{\eta+1}} & \text{if } \mu \le 0.5 \\ \left[\frac{1}{2(1-\mu)}\right]^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases} \quad (7)$$

After obtaining β_q from the above probability distribution, the children solutions are calculated as follows:

$$Child_i^1 = 0.5[(1+\beta_q)\times Parent_i^1 + (1-\beta_q)\times Parent_i^2 \quad (8)$$

$$Child_i^2 = 0.5[(1+\beta_q)\times Parent_i^2 + (1-\beta_q)\times Parent_i^1 \quad (9)$$

Linear BGA crossover: The objective is to generate offspring better than its parents, linear BGA crossover approach is to create offspring closer the best parent (Muhlenbein and Schlierkamp-Voosen, 1993). This operator not carry out a sampling around the regions of parents but it takes into account the parents' fitness.

Two parents $Parent_i^1$ and $Parent_i^2$ generate $Child_i^1$. Let, $Parent_i^1$ be the parent with better fitness. Accordingly, the offspring has genes calculated in the following way:

$$Child_i^1 = Parent_i^1 \pm r_i\times\lambda\times\gamma \quad (10)$$

where, λ is calculated according to:

$$\lambda = \left|\frac{Parent_i^2 - Parent_i^1}{Parent^2 - Parent^1}\right| \quad (11)$$

where, the minus sign is chosen with a probability of 0.9 and $r_i = 0.5(a_i-b_i)$. It should also be noted that each gene $Parent_i^2\in[a_i, b_i]$ and γ equals to:

$$\gamma = \sum_{k=0}^{15} \alpha_k 2^{-k} \quad (12)$$

where, $\alpha_k\in[0, 1]$ is generated with random manner.

MATERIALS AND METHODS

The proposed crossover: After implementing the four crossover operators described in the previous study and tested them on the optimization problem of a variety of test functions we found that results differ significantly from one operator to another. This poses the problem of selecting the adequate operator for real-world problems for which no posterior verification of results is possible.

To help mitigate this non-trivial problem we present in this section the outlines of a new crossover that we propose as an alternative which can be useful when no single other technique can be used with enough confidence. The technique is a dynamic one in the sense that the crossover can vary from one generation to another. The underlying idea consists in finding a good compromise between assuring some genetic diversity within the population but may increase the convergence time and producing best style of offspring that reduce the convergence time but a possible risk of converging to local minima.

To achieve this goal, more than one crossover operator are applied at each generation but in a competitive way meaning that only results provided by the operator with the best performance are actually taken into account. To assess and compare the performance of candidate operators two objective criteria are employed. The first criterion is the quality of solution; it can easily be measured as a function of the fitness of the best individual. The second criterion is the manner to create the best style of offspring which is less evident to quantify than the first one. Radcliffe (1993) proposed six criteria for the best performance of crossover operator but he has not suggested any comparison measure. Deb proposed the spread factor of offspring conformed to the parent, this measure provided the crossover ability to create any random point in the search space. But the distribution of the spread factor as a function of the crossover varies in the same direction of the parent. In this work, we reformulate this factor taking into account the convergence of a population. This convergence represents the dynamics of grouping the individuals around the optimum. To measure this characteristic, we introduce the Euclidean distance between the barycentre and the individual of the population:

$$O_p = \frac{\sum_{i=1}^{N} P_i}{N} \tag{13}$$

Where:

P_i = An individual
O_p = The barycenter of all individuals
N = The population size

The modified spread factor become:

$$O_p = \frac{\text{Max}(C_1\text{-}O_p,\ C_2\text{-}O_p)}{\text{Max}(P_1\text{-}O_p,\ P_2\text{-}O_p)} \tag{14}$$

As a measure of the quality of the solution at each generation, we used the following criterion:

$$O_q = \frac{\sqrt{2}f^*}{\sqrt{f_{max}^2 + f_{min}^2}} \tag{15}$$

where, f_{max} and f_{min} denote respectively the maximum and the minimum values of the fitness at generation t and $f_{max} = f^*$ or $f_{min} = f^*$ depending on the nature of the problem which can be either a maximization or a minimization problem. Finally, in order to combine the two criteria in a unique one we used the relation:

$$O_f = \frac{1}{t}O_\beta + \frac{t\text{-}1}{t}O_q \tag{16}$$

RESULTS AND DISCUSSION

The proposed method is tested by a six functions widely used in performance evaluation of GA operators. Benchmark functions used in this paper have two important features: modality and separability. Uni-modal function is a function with only one global optimum. The function is multi-modal if it has two or more local optima. Multi-modal functions are more difficult to optimize compared to uni-modal functions.

Description: Ackley problem, n = 5 multi-model:

$$f_1(\bar{x}) = -20.e^{\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)} - e^{\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)} + 20 + e$$

$$-30 \le x_i \le 30,\ x^* = (0, 0, ..., 0),\ f(x^*) = 0$$

Cosinus mixture problem, n = 10:

$$f_2(\bar{x}) = \sum_{i=1}^{n} x_i^2 - 0.1\sum_{i=1}^{n}\cos(5\pi x_i)$$

With:

$$-1 \le x_i \le 1,\ x^* = (0, 0, ..., 0),\ f(x^*) = 0.1 \times n$$

Goldstain-price problem, n = 2:

$$f_3(\bar{x}) = \left[1+(x_0+x_1+1)^2\left(19-14x_0+3x_0^2-14x_1+6x_0x_1+3x_1^2\right)\right]\times \left[30+(2x_0-3x_1)^2\left(18-32x_0+12x_0^2+48x_1-36x_0x_1+27x_1^2\right)\right]$$

$$-2 \le x_i \le 2,\ x^* = (0, -1),\ f(x^*) = 3$$

Griewank problem, n = 5:

$$f_4(\bar{x}) = 1+\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$$

$$-600 \le x_i \le 600,\ x^* = (0, 0,..., 0),\ f(x^*) = 0$$

Levy et montalvo problem 1, n = 5:

$$f_5(\bar{x}) = \frac{\pi}{n}\left(10\sin^2(\pi y_1)+\sum_{i=1}^{n-1} g(y_i)\right)$$
$$g(y_i) = (y_i-1)^2\left[1+10\sin^2(\pi y_i+1)\right]+(y_n-1)^2$$

$$y_i = 1+\frac{1}{4}(x_i+1),\ -5\le x_i \le 5,\ x^* = (-1, -1, ..., -1),\ f(x^*) = 0$$

Paviani problem, n = 10:

$$f_6(\bar{x}) = \sum_{i=1}^{10}\left[(\ln(x_i-2))^2+(\ln(10-x_i))^2\right]+\left(\prod_{i=1}^{10} x_i\right)^{0.2}$$

$$2 \le x_i \le 10,\ x^* \approx (9.350266, 9.350266,..., 9.350266)$$

$$f(x^*) = -45.778470$$

In all experiments, the stochastic uniform selection was used. Parameters of GA for experiments were as following.

Gaussian mutation with p_m mutation rate equal to 0.01 and crossover rate p_m varies for each run, a number of independent runs for each experiment was 30, initial population N of size 100 was randomly created and used in experiments. Dimensionality of the search space D for all test functions varie between 2 and 10. Number of overall evaluations were set to 10000. For all test functions, finding global minimum is the objective. All of the experiment are realized for six different types of test functions. A comparison between the proposed Crossover Method (CCO) and other crossover methods in terms of the quality assessment of the optimum provided by the GA are made and the results are comparatively presented in Table 1. To measure this quality, we used the relative error:

$$R_e = \left|\frac{f^*-f}{f}\right| \quad (17)$$

Where:

f^* = The optimum provided by the algorithm
f = The actual optimum which is a priori known

Table 1: Relative error in percentage of the optima

Test functions	Heuristic	Arithmetic	SBX	BGA	CCO
F1	18	7	7	5	4
F2	5	5	6	5	3
F3	10	15	15	10	8
F4	10	15	10	16	3
F5	25	18	21	22	9
F6	20	15	15	2	1

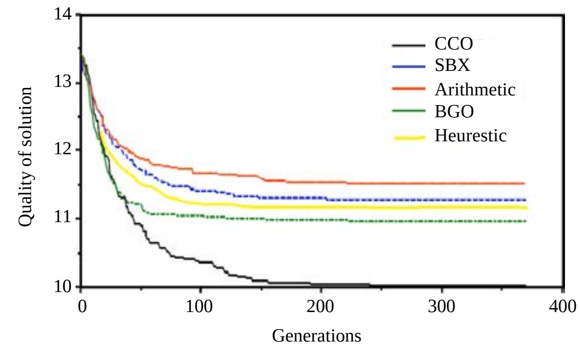

Fig. 1: Convergence plot of the function F6

CCO gives better results according to other crossover operators. Although, the most of crossover operators showed similar results, CCO had slightly better results than the other crossover for F1, F2, F4, F5 functions. For F3 function, SBX operator has slightly near the result than CCO. However, CCO operator produces better results than other crossover operators. For F6 function, the results of this study are very close to those of BGA but in generally CCO operator performed better results than other crossover operators. The most important advantage of the proposed method is that more variety is presented in possible number of children according to heuristic crossover and linear BGA crossover. The experiments and the results presented in the study clearly reveal the potential capability of the proposed method in optimization processing based on GA.

Figure 1 shows the convergence plots obtained by minimizing the test function F6 it is clear that CCO tends to effectively exploit the search space, this is inferred by the number of generations needed to find the optimum.

CONCLUSION

In this study, four well-known crossover operators for GAs are studied, implemented and their relative performance analysed and compared using a set of six well-known test functions. These operators can be tested with two criteria: the quality of the solution and the spread of the distribution. The first one marks the performance of the crossover operator, the second shows the spread of population distribution and the ability of GA for not to be trapped in the local optima.

By contrast, the crossover operator proposed exploits the advantages of each crossover used over the evolution of the GA. The main idea behind this operator is the use of more than one crossover operator in a competitive way together with an objective criterion which allows choosing the best operator to adopt at each generation. The proposed technique was successfully applied to the optimisation problem of a set of well-known test functions which encourages further developments of this idea.

REFERENCES

Deb, K. and A. Kumar, 1995. Real-coded genetic algorithms with simulated binary crossover: Studies on multimodel and multiobjective problems. Complex Syst., 9: 431-454.

Deb, K. and R.B. Agrawal, 1995. Simulated binary crossover for continuous search space. Complex Syst., 9: 115-148.

Deep, K. and M. Thakur, 2007. A new crossover operator for real coded genetic algorithms. Applied Math. Comput., 188: 895-911.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. 1st Edn., Addison-Wesley Professional, Boston, MA., USA., ISBN-13: 9780201157673, Pages: 412.

Herrera, F. and M. Lozano, 1996. Heuristic Crossovers for Real-Coded Genetic Algorithms based on Fuzzy Connectives. In: Parallel Problem Solving from Nature-PPSN IV, Voigt, H.M., W. Ebeling, I. Rechenberg and H.P. Schwefel (Eds.). Springer-Verlag, Berlin, Germany, ISBN: 978-3-540-61723-5, pp: 336-345.

Herrera, F., M. Lozano and A. M. Sanchez, 2003. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. Int. J. Intell. Syst., 18: 309-338.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. 1st Edn., University of Michigan Press, Ann Arbor, MI., USA., ISBN-13: 9780472084609, Pages: 183.

Hong, T.P., H.S. Wang, W.Y. Lin and W.Y. Lee, 2002. Evolution of appropriate crossover and mutation operators in a genetic process. Applied Intell., 16: 7-17.

Lobo, F.G., C. F. Lima and Z. Michalewicz, 2007. Parameter Setting in Evolutionary Algorithms. Springer, New York, USA., ISBN-13: 9783540694311, Pages: 317.

Muhlenbein, H. and D. Schlierkamp-Voosen, 1993. Predictive models for the breeder genetic algorithm. I. Continuous parameter optimization. Evol. Comput., 1: 25-49.

Ortiz-Boyer, D., C. Hervas-Martinez and N. Garcia-Pedrajas, 2005. CIXL2: A crossover operator for evolutionary algorithms based on population features. J. Artif. Intell. Res., 24: 1-48.

Poli, R. and W.B. Langdon, 2006. Backward-chaining evolutionary algorithms. Artif. Intell., 170: 953-982.

Radcliffe, N., 1993. Genetic Set Recombination. In: Foundations of Genetic Algorithms 2, Whitley, L.D. (Ed.). Morgan Kaufmann, San Mateo, CA., USA., ISBN-13: 9781558602632, pp: 203-219.

Reeves, C.R. and J.E. Rowe, 2002. Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory. Kluwer Academic Publishers, Norwell, MA., USA., ISBN-13: 978-1402072406, Pages: 332.

Zbigniew, M., 1996. Genetic Algorithms + Data Structures = Evolution Programs. 3rd Rev. Edn., Springer, Berlin, Germany, ISBN-13: 9783540606765, Pages: 387.