THE MCPC EVOLUTION IN EVOLUTIONARY COMPUTATION

Esquivel S. C., Leiva H. A., Gallard R. H.
Proyecto UNSL-338403[1]
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
5700 - San Luis, Argentina.
E-mail: {esquivel, aleiva, rgallard}@unsl.edu.ar
Phone: +54 2652 420823
Fax : +54 2652 430224

Abstract

The alternative proposed and known as MCPC [6] has improved the performance of the original Holland's Genetic Algorithms in order to obtain good quality solutions in less execution time. But, it's brought out an important concept in Evolutionary Computation: the relationship between exploration and exploitation of solutions in a problem space.

The current presentation briefly describes the improvements of the MCPC approach to face the balance between exploration and exploitation.

1. Introduction

Important research has been made in evolutionary computation, to maintain a good balance between exploration and exploitation of solutions in a problem space. This research has involved the study of the effect of selection mechanisms. Strong selective pressure can lead to premature convergence towards local optima, while the opposite can make the search ineffective because it would take long time to reach some individual near optimum [11].

Also, the recombination stage of an Evolutionary Algorithms (EA) has its own contribution to the search process. A low recombination rate can impede binary schema processing permitting super-individuals to cope the population and leading to premature convergence. On the other hand, a high rate can be, in some cases, too disruptive allowing the loss of good genetic material, slowing and spoiling the search.

The intuition behind the applicability of the crossover operator is information exchange between different potential solutions. The common approach is to operate once on each mating pair after selection. There, we call such a procedure SCPC (*Single Crossover Per Couple*) approach. Mimicking, in some degree, what happens in nature we devised a different approach to allow multiple offspring per couple and called it MCPC (*Multiple Crossovers per Couple*)[6].

MCPC tries to promote the exploitation of good previously found solutions favouring, with more copies, the best individuals in the current population. We achieved this by repeatedly applying the crossover method to the selected mating pairs; so each of them generates multiple children.

The idea of multiple children per couple was tested on a set of well-known testing functions (De Jong functions F_1, F_2 and F_3[1], Schaffer F_6 [12] and other functions). A simple genetic algorithm, with conventional operators and parameter values, was the basis of those initial experiments. Allowing multiple crossing between selected parents similar and better quality solutions were obtained when contrasted against the conventional crossover approach (SCPC). Also, a deeper

---

[1]The Research Group is supported by the Universidad Nacional de San Luis and the CONICET (Science and Technology Research Council).

decrease on running time was observed as long as the number of crossover per couple increased. This was due to a lesser number of applications of the selection method. But, on the other hand, this approach showed, in some cases, a low exploration level because only a small portion of the actual population effectively undergoes recombination, increasing the risk of premature convergence due to a loss of genetic diversity.

To overcome this problem further successful approaches were undertook by combining MCPC with an alternative selection mechanism called *fitness proportional couple selection* (FPCS) [7], using self-adaptation of MCPC parameters [8], and by binding MCPC to alternative selection mechanisms [9].

Two of the more important approaches, based on multiplicity, that have been designed and implemented to improve the searching process are MCPC and *multiparent recombination*.

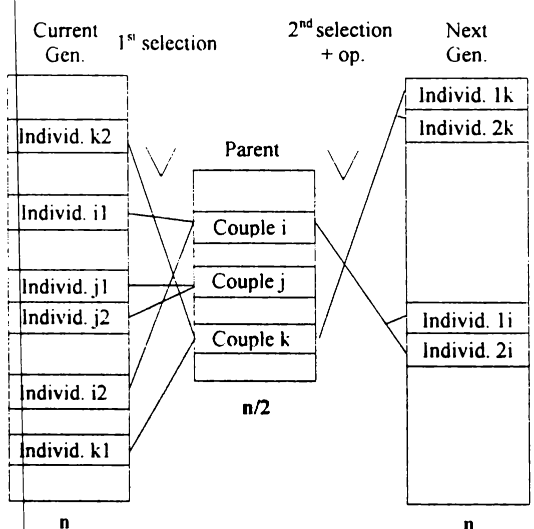
Further investigation was addressed to combine both multiplicity approaches [10]. Consequently MCPC was naturally extended to *multiple crossovers per mating* (MCPM). In his original work Eiben [2], [3], [4], [5] proposed uniform scanning crossover, a relatively new crossover method. Here, from multiple parents, a single offspring is generated where each gene is provided from any of the corresponding genes in the parents with equal probability.

The current presentation will briefly describe the evolution of the MCPC approach incarnated by the sequence of the above mentioned methods and will give insights on the performance of evolutionary algorithms optimizing two highly multimodal (Griewank's and Schweffel F7 [14]) functions and a hard unimodal (Easom's) function [13].

2. Splitting the selection process

2.1. The FPCS method

FPCS divides the whole selection process into two stages. The first one selects individuals from the current population to create an intermediate group of couples, and subsequently the second stage undergoes this population of couples to selection for mating those pairs showing higher fitness (see Fig. 1). Both selection steps are based on the traditional proportional selection scheme.



1st Selection: proportional selection to the individual fitness.
2nd selection + op: proportional selection to the couple fitness plus classic genetic operators.
n: population size.

Fig. 1. Couple Selection Process Scheme

The selection operator can be modified to favour, with more children, those couples having specific characteristics. There are no problems about individual fitness but two criteria to assign fitness to couples were chosen; one based on members fitness dissimilarity ($FPCS_{DIFF}$) and the other on their average fitness ($FPCS_{AVG}$).

The method used can be sketched as follows;

- A number of individuals are initially selected by proportional selection to build the intermediate population of parents.

- A couple fitness value, computed in accordance to the couple fitness criterion, is assigned to each mating pair.

- Couples are selected for reproduction by proportional selection (according to couple fitness). The process of producing offspring is controlled, for each mating pair, in order to not exceed the population size.

2.2. Using MCPC with FPCS

Looking for improvements in the genetic algorithm performance regarding to execution time, exploitation and exploration of the searching space, we combined the two techniques above-mentioned.

For this approach we show results on those functions that showed to be more difficult to optimise for the simple (modified) GA. They were:

- the *Griewank's function 8*, called by us *Griewank's function* simply, and
- the *Easom's function*

The algorithm was performed on every function many times, they were series of 6 runs each one. The first run corresponds to SCPC and the others to MCPC, because it was fixed that the *i* run performed *i* crossovers per couple exactly, that means *2i* children per each of them. So, the number of children per couple was fixed in each run and the process of producing new individuals was controlled in order not to exceed the population size: in every case each new created offspring was inserted into the next generation until the population size was reached.

The experiments were based on a simple, but not canonical, Genetic Algorithm with binary coded chromosomes, elitism, bit swap mutation, one point crossover, population size fixed to 70 individuals and probabilities of 0.5 for crossover and 0.005 for mutation. Except for the Easom's function where the crossover and mutation probability were set to 0.65 and 0.05 respectively. The number of generations allowed ranged from 2500 to 5000.

After experimental runs the following relevant performance variables were examined:

Ebest: It is the percentile error of the best-found individual value when compared with the known, or estimated, optimum value. It gives a measure of how far the best individual is from the optimum point.

Epop: It is the percentile error of the population mean fitness when compared against the optimum value. It shows how far the mean fitness is from the optimum value.

Gbest: Indicates the generation where the fittest individual (retained by elitism) was found.

All the values analysed were mean values obtained from the series completed for each fixed number of crossovers, on each function.

2.3. Results

Combining both techniques good results were obtained for all optimised functions when they were contrasted against those results showed by MCPC approach alone [7]. In Tables 1 and 2 we summarize the most relevant values obtained by $FPCS_{AVG}$.

Performance variable	Minimum value	Maximum value
Ebest	0.0381	0.0836
Epop	7.1384	9.8447
Gbest	2026	4017

Table 1– Performance variables values for Griewank's function

Performance variable	Minimum value	Maximum value
Ebest	0.0124	2.5176
Epop	44.9970	47.6402
Gbest	839	2340

Table 2– Performance variables values for Easom's function

3. Finding a suitable crossover number

3.1. Implementation

From previous work using MCPC combined with others techniques we established that good quality results were obtained allowing between 2 and 4 crossovers per couple. But setting an adequate number of crossovers per couple is a hard task.

As an attempt to find this optimum number, was to codify the number of crossovers inside each individual of the population. So the GA will self-adapt this parameter and give the most appropriate quantity of crossover repetition. Self-adaptation of parameters is one of the three techniques mentioned in [16] to dynamically change a parameter value while the algorithm is running.

We achieved this self-adaptation codification using the last $log2(max\_cross + 1)$ bits of each individual to find the expected optimum. These last bits are called *ncross_field*.

In that way we have two searching spaces: one corresponding to the objective function and other associated to the number of crossovers to apply.

Each individual preserves information about the number of crossovers originally applied to their parents. In this way it is expected that, based on the *survival-of-the-fittest* principle, good solutions carry information about the number of crossover applied to their ancestors and that this number would be an appropriate one.

According to Spears [17] we used a local self-adaptive technique. Once the couple was selected we checked the corresponding number of crossover carried by each parent and;

- If they match, then we apply the recombination operator a number of times specified by the *ncross_field*.
- Otherwise we use a random number in the permitted range instead of *ncross_field* number.

In the second situation and following the Spears's approach, when decoded numbers of crossovers are different, we would be violating our attempt to preserve information because the children will not keep the number of crossover by which they were created. If crossover and mutation do not disrupt the *ncross_field* (and this event has low probability to occurs) then the children retain information from either parent, but they do not preserve information about how they were created.

In order to either retain information about how an individual was created or how their parents were created, we devised two different approaches for experimentation.

- *E1*: In any situation, exchange of information from parents to children is done in the traditional way applying the genetic operators with their corresponding probabilities. In the *don't match* situation, this approach, preserving parent's information, enforces population diversity in the parameter searching space, because most of the time one child inherits characteristics from one parent and the other child inherits features from the other parent.
- *E2*: If the values specified in *ncross_field* do not match then the new random value for the crossover number is inserted first in the parent's *ncross_field,* and afterwards crossover is performed for the number of times specified by this random value. This approach by preserving individual information creates more similar individuals in the parameter searching space and increases loss of genetic diversity.

3.2. Empirical tests

We analyzed the self-adaptation of the crossovers number from the results obtained from two hard testing functions: the above cited unimodal Easom's function and Schweffel's function 7, a multimodal one.

Here, a non canonical GA similar to the preceding one and with the same main parameter settings was used. Except, of course, a series does not consist of 6 runs anymore because there is only one run that attempts to find the most appropriate number. Also, the generation number was fixed to

500 and probabilities to apply crossover and mutation at 0.65 and 0.05, respectively for both functions.

Several experiments were conducted to evaluate the behavior of E1 and E2 experimental groups. Some of them used two bits to code *ncross_field* and the remaining used three. So, we allowed up to three crossovers per couple in some cases and seven in the others.

After running the experiments many times we were interested in a performance variable subject of this study called *Cr_Avg* that measures the mean crossover number allowed per couple, and in the *Quality* of the solutions obtained [8].

In both cases *Quality* reached the value 1 (at least one solution was optimal) after 80 and 152 generations respectively and *CrAvg* values ranged between 2.5 and 2.8 after a few generations under E1 and between 2.3 and 2.6 under E2.

Here the behavior of the self adaptive parameter control mechanism is clear: when genetic diversity in the parameter searching space is low then lesser number of crossovers are allowed and viceversa. This behavior favours the evolutionary process.

4. Alternative selection mechanisms

Attempting to dismiss the selective pressure introduced by applying jointly MCPC and selection methods based on the individual or couple fitness proportion we replaced proportional selection by selections based on linear ranking schemes. Also, in these experiments we used another of the three techniques mentioned in [16] to adapt the η_{max} ranking selection parameter: the deterministic one, that changes the parameter value according to a deterministic rule, without any feedback of the searching process performed by the strategy.

One of the main purposes of dynamically change a parameter value is that the algorithm adequately tunes to the best setting for the particular problem that is solving. Particularly, when ranking selection is used, it is not an easy task to tune the expected value of the number of offspring for the best individual: η_{max} . This parameter highly influences selective pressure.

In this work we studied the effect of MCPC when it was jointly applied to *static ranking selection* (SRS) and *deterministic dynamic ranking selection* (DDRS) in order to moderate the joint effect of Proportional Selection (PS) and MCPC.

DDRS, is proposed as a selection method which updates deterministically and dynamically the above –mentioned parameter as a function of the number of generations reached. In this case η_{max} is given by the following expression:

$$\eta_{max} = (\#current\_gen + \#max\_gen) / \#max\_gen$$

By using this variant of ranking we attempt to enforce exploration during the earlier stages and exploitation during the final stages of the evolution process. At the beginning, selective pressure is weak and increases smoothly through the iterations reaching the maximum selective pressure allowed by ranking at the end of the process. In this way we can expect to slow the convergence rate to prevent being trapped in local optima.

4.1. Experimental tests and results

Experiments were designed to compare results when optimising the last two mentioned functions using MCPC and SCPC, under PS, SRS and DDRS. For SRS two values of η_{max} were considered, $\eta_{max} = 1.2$ (low selective pressure) and $\eta_{max} = 1.6$ (intermediate selective pressure).

Many series of 6 runs each one using the same parameters settings as the above experiments about self-adaptation were performed on each functions.

[illegible] (above defined) was one of the central performance variables studied. In the case of the hard multimodal (Schwefel's) function it was observed that except for SRS (1.2) all other methods found the optimum value for some number of crossovers before the end of the simulation. PS resulted in general, more efficient in this case. Moderate SRS (1.6) is the best when four crossovers are allowed. DDRS, effectively tuned the selective pressure to low values when exploration is needed and gradually increments it to higher values as simulation time progresses. This effect is necessary to avoid premature convergence in complex multimodal fitness landscapes. In the case of the hard unimodal (Easom's) function it was observed that when any ranking method is used then SCPC improves its own results previously obtained under PS. Also, with diverse convergence speeds all the ranking selection mechanisms found the optimum for any number of crossovers. Former studies with similar parameter settings, under PS, were unsuccessful in finding the optimum.

Summarizing, the use of rank-based selection methods is in general better than PS when combined with MCPC, especially when they are used to optimise functions with multiple local minima.

5. Multiple Crossover on Multiple Parents (MCMP)

This new method allows multiple parents to be recombined several times. Each time, under uniform scanning crossover, they generate only one child as it was mentioned in the introduction. Once selected, the n_1 parents undergo crossover a number n_2 of times specified as an argument and generates n_2 children, subsequently the fittest child is selected for insertion in the next generation.

In our extension of MCPC, when multiple parents are selected based on their fitness, MCPM provides a means to exploit their good features. Also, as long as we permit a greater number of parents to take part of an offspring creation a larger sample from the searching space is considered and consequently larger diversity is supplied obtaining a greater exploration. So a good balance between exploration and exploitation of solutions was achieved as results show.

Summary of relevant results can be seen in tables 3 and 4. The characteristics of the GA and the functions used here were the same as the first method listed at the beginning, except that the maximum generation number was decreased to 500. When n_1 parents are selected by fitness proportional selection, then they undergo to MCPM and bit swap mutation to obtain n_2 offspring. Subsequently, the fittest child was selected for insertion in the next generation.

Performance variable	Minimum value	Maximum value
Ebest	0.0074	0.4660
Epop	0.0074	0.1694
Gbest	73	491

Table 3 – Performance variables values for *Griewank's function*

Performance variable	Minimum value	Maximum value
Ebest	0.0000	0.0415
Epop	0.0798	9.9710
Gbest	43	400

Table 4– Performance variables values for *Easom's function*

Both tables show results that outperform previous results under MCPC and FPCS, detailed in tables 1 and 2, and are of similar quality of those obtained on the Easom's function by self adaptation of MCPC parameters and by joining MCPC and adaptive ranking selection (sections 3 and 4). An important characteristic of the method is the ability to provide a final population densely grouped around the optimum. This effect was not perceived in previous experiments.

Moreover, varying n_1 from 3 to 8 and n_2 from 1 to 4 it was observed better results for the MCMP approach when contrasted against the original multiparent approach with single crossover per mating. These full results are reported in [10].

14. Summary and conclusions

By applying MCPC technique alone good solutions were obtained for a widespread range of functions. Also, because the method selects a lesser number of mating pairs, the GA execution time was reduced. But the homogeneity of this method jointly to the inherent strong selective pressure of proportional selection, sometimes, caused the search be trapped in local optima. So we tried to do better through the hybrid techniques above revisited.

Each techniques attempted to face particular weaknesses of MCPC. Therefore, FPCS tried to do a more exhaustive search across the space, MCPC parameters self-adaptation addressed to find the most appropriate number of crossover been applied to a couple, the use of ranking selection schemes dismissed the stronger selective pressure (introduced by joining MCPC and proportional selection) and finally the use of multiple recombination on multiple parents showed to be efficient in optimisation of hard unimodal and multimodal testing functions. It seems that the multiparent approach mitigates the possible loss of diversity generated by multiple crossover and no extra adjustments, used before, seem to be necessary. Consequently the quality of results is at least as good as previous more complex approaches. Additionally, when observing the final population it was detected that all individuals are much more centred surrounding the optimum. This is an important issue when the application requires provision of multiple alternative near-optimal solutions.

Although we cannot be conclusive, it seems that by means of this association the searching space is efficiently exploited by the multiple application of crossovers and efficiently explored by the greater number of samples provided by the multiple parents.

In view of these promising results new work is done to study the effect of multiple crossovers on multiple parents under diverse crossover methods.

7. Bibliography

[1] De Jong K. A., Analysis of the Behavior of a Class of Genetic Adaptive Systems, PhD Dissertation, University of Michigan, 1975.

[2] Eiben A. E., Raué P. E., and Ruttkay Zs.: *Genetic algorithms with multi-parent recombination.* In Davidor, H.-P. Schwefel, and R. Männer editors, Proceedings of the 3rd Conference on Parallel Problem Solving from Nature, number 866 in LNCS, pages 78-87. Springer-Verlag, 1994

[3] Eiben A.E., van Kemenade C.H.M., and Kok J.N.: *Orgy in the computer: Multi-parent reproduction in genetic algorithms.* In F. Moran, A. Moreno, J.J. Merelo, and P. Chacon, editors, Proceedings of the 3rd European Conference on Artificial Life, number 929 in LNAI, pages 934-945. Springer-Verlag, 1995.

[4] Eiben A.E. and. Bäck Th., *An empirical investigation of multi-parent recombination operators in evolution strategies.* Evolutionary Computation, 5(3):347-365, 1997.

[5] Eiben A.E. and van Kemenade C.H.M., *Diagonal crossover in genetic algorithms for numerical optimization.* Journal of Control and Cybernetics, 26(3):447-465, 1997.

[6] Esquivel S., Leiva A., Gallard R.: *Multiple Crossover per Couple in Genetic Algorithms.* Proceedings of the Fourth IEEE Conference on Evolutionary Computation (ICEC'97), Indianapolis, USA, April 1997.

[7] Esquivel S., Leiva A., Gallard R.: *Couple Fitness Based Selection with Multiple Crossover per Couple in Genetic Algorithms.* Proceedings of the International Symposium on

Engineering of Intelligent Systems (EIS'98), La Laguna, Tenerife. Spain. ed. E.Alpaydin. Publishesd by ICSC Academic Press. Canada/Switzerland. February 1998.

[8] Esquivel S., Leiva H., Gallard R.: *Self-Adaptation of Parameters for MCPC in Genetic Algorithms*. Proceedings of the 4th Congreso Argentino de Ciencias de la Computación (CACiC'98). Universidad Nacional del Comahue. Argentina. October. 1998.

[9] Esquivel S., Leiva H., Gallard R.: *A Study of Alternative Selection Mechanisms for Multiple Crossover per Couple in Genetic Algorithms*. Proceedings of the 4th Congreso Argentino de Ciencias de la Computación (CACiC'98). Universidad Nacional del Comahue. Argentina. October 1998.

[10] Esquivel S., Leiva H., Gallard R.: *Multiple crossovers between multiple parents to improve search in evolutionary algorithms,* accepted for publication in the Proceeding of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC.

[11] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, third revised edition, 1996.

[12] Schaffer J., Caruana R., Eshelman R., Das R.: *A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization*, in Third International Conference on Genetic Algorithms, 1989.

[13] Easom, E.: *A survey of global optimization techniques*. M. Eng. Thesis, Univ. Louisville, Louisville, KY, 1990. In [15].

[14] Schwefel, H. P.: *Numerical optimization of computer models*. Chichester: Wiley & Sons, 1981. In [15].

[15] Hartmut Pohlheim: *Genetic and Evolutionary Algorithm Toolbox for use with Matlab (GEATbx)*. Copyright © 1996, Germany. All rights reserved.

[16] Eiben A. E., Hinterding R., Michalewicz Z.: *Parameter Control in Evolutionary Algorithms*. Technical Report, UNC - Charlotte, 1998.

[17] William M. Spears: *Adapting Crossover in Evolutionary Algorithms*. Proceedings of the Evolutionary Programming Conference, 1995.

8. Appendix

Description of the functions optimized by the methods above listed:

Notation	Description		Characteristics
Easom's Function	$f(x_1, x_2) = -\cos(x_1)\cos(x_2)\, e^{-((x_1-\pi)^2+(x_2-\pi)^2)}$, $x_1, x_2 \in [-100, 100]$		Unimodal, the global minimum has a small area relative to the search space
Griewank's Function	$f(x_i \mid i=1,5) = 1 + \sum_{i=1}^{5} \frac{x_i^2}{4000} - \prod_{i=1}^{5}\left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)$, $x_i \in [-600, 600]$	Dim. used $n = 5$	Multimodal, however, the local minima are regularly distributed.
Schwefel's Function 7	$f(\vec{x}) = \sum_i -x_i \cdot \sin\left(\sqrt{\lvert x_i \rvert}\right)$ *for* $i = 1:n$, $x_i \in [-500, 500]$	Dim. used $n = 5$	Highly multimodal, the global minimum is geometrically distant from the next best local minima.