ENHANCING EVOLUTIONARY ALGORITHMS THROUGH RECOMBINATION AND PARALLELISM

GALLARD R.H., ESQUIVEL S. C.

Proyecto UNSL-338403[1]
Departamento de Informática
Universidad Nacional de San Luis (UNSL)
Ejército de los Andes 950 - Local 106
5700 - San Luis, Argentina.
E-mail: { rgallard esquivel}@unsl.edu.ar
Phone: + 54 652 20823
Fax : +54 652 30224

Abstract

Evolutionary computation (EC) has been recently recognized as a research field, which studies a new type of algorithms: Evolutionary Algorithms (EAs). These algorithms process populations of solutions as opposed to most traditional approaches which improve a single solution. All these algorithms share common features: reproduction, random variation, competition and selection of individuals. During our research it was evident that some components of EAs should be re-examined. Hence, specific topics such as multiple crossovers per couple and its enhancements, multiplicity of parents and crossovers and their application to single and multiple criteria optimization problems, adaptability, and parallel genetic algorithms, were proposed and investigated carefully. This paper show the most relevant and recent enhancements on recombination for a genetic-algorithm-based EA and migration control strategies for parallel genetic algorithms. Details of implementation and results are discussed.

Keywords: Evolutionary algorithms, multirecombination, strategies for migration control.

[1] The Research Group is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).

1. INTRODUCTION

In evolutionary algorithms all conventional approaches apply the crossover operator only once on the selected parents. But in nature when the mating process is carried out, crossover is applied many times and the consequence is a multiple and variable number of offspring. The question arising is: how would the performance of an EA be affected by the use of a multiple crossovers per couple (MCPC) operation? Exploration and exploitation of solutions in the searching space are distinctive characteristics of an evolutionary algorithm, and are responsible for the success or failure of the search process. Extreme exploitation can lead to premature convergence and intense exploration can make the search ineffective. To find a balance between these two factors is of paramount importance for the EA performance when speed of the search and quality of results are involved. Many researchers focus on the balancing problem studying the effect of selection mechanisms, because selective pressure can adjust exploration and exploitation. On its own, recombination can also participate on this respect but depending on how it is applied it can aid or disrupt the search process. For example, a low rate for recombination can impede schema processing permitting super-individuals to replenish the population, thus leading to premature convergence. On the other hand a high rate can be, in some cases, extremely disruptive allowing good genetic material to be lost, slowing down the search.

Parallel implementations of Genetic Algorithms (GAs) also aim at improvements on performance. The main purpose of this approach is to enhance the quality of the results. The *island model* [2], [3], [18], [19], a well known distributed approach, where separate subpopulations evolve in parallel is a realistic model of natural evolution which is appropriate for a distributed environment running a Single Program Multiple Data (SPMD) scheme.

The following sections discuss new approaches to enhance EAs performance via multirecombination and parallelism, and show some results.

2. A MULTIPLICITY FEATURE OF EVOLUTIONARY ALGORITHMS

This is the main contribution of this work in the theoretical field of Evolutionary Computation. The multiplicity feature is related to new proposed multi-recombination methods:

- MCPC: Multiple Crossovers per Couple which reinforces the exploitation of features of previously found (good) solutions.
- MCMP: Multiple Crossovers on Multiple Parents which provides a balance in exploitation and exploration because the searching space is efficiently exploited (by the multiple application of crossovers) and explored (by a greater number of samples provided by multiple parents).

The multiplicity feature was tested in the optimization of hard testing functions: Griewank’s, Schaffer´s F_6, and F_7, Shubert´s (highly multimodal functions), Easom’s and the Volcano [4] (difficult unimodal functions).

2.1. MCPC AND ITS ENHANCEMENTS

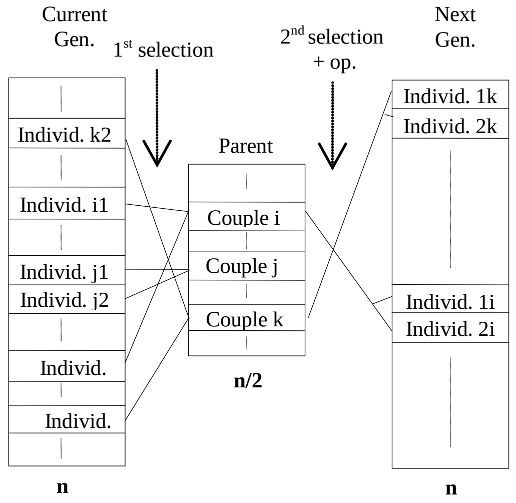
The crossover operator provides a major contribution to the process of exchanging genetic material during the execution of an EA. Conventional crossover combines the features of two parent chromosomes to form two similar offspring by swapping corresponding segments of the parents. The intuition behind the applicability of the crossover operator is information exchange between different potential solutions. The common approach to crossover is to operate once on each mating pair after selection. From now on such procedure will be called the *single crossover per couple* (SCPC) approach. We devised a different approach to allow multiple offspring per couple, to explore the recombination possibilities of previously found solutions. In our earlier works [6], [7], a simplified version of MCPC was used. During those first studies of the MCPC approach it was observed that:

- In some cases this simple MCPC method found results that were better than those found by the SCPC method.
- Running time improved as long as the number of crossover per couple increased.
- Best quality results were obtained allowing between 2 and 4 crossovers per couple.
- In some cases, the method increased the risk of premature convergence due to a loss of genetic diversity.

These effects were a consequence of saving computational effort and of a greater exploitation of the recombination of good, previously found solutions. To overcome the premature convergence problem, further successful approaches were undertaken by combining MCPC with an alternative selection method: *fitness proportional couple selection* (FPCS) [8], by using self-adaptation of MCPC parameters [9], by binding MCPC to alternative selection mechanisms [10] or by allowing multiple parents and crossovers [11]. All these approaches outperformed the original MCPC approach, at higher but no sensitive computational cost. They are briefly described now.

2.1.1. MCPC WITH FPCS

Depicted in figure 1, the method can be sketched as follows.



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

- A number of individuals are initially selected by proportional selection to build the intermediate population of parents, which are grouped randomly into pairs.
- 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 not to exceed the population size.
- The number of offspring per couple is assigned by means of a mapping between the couple fitness values in the current population (which are grouped into as many ranges as number of crossovers are there to be applied) and the possible number of crossovers (which ranges from one to the maximum number allowed).

2.1.2. SELF ADAPTATION OF MCPC PARAMETERS

This approach attempts to self-adapt the number of crossovers per couple in MCPC. Because we are using a binary representation of chromosomes, the number of crossovers allowed for an individual is codified in a field at the rightmost positions of the bit string. Let us call it the *ncross_field*. In some experiments we allowed a maximum of three and in others a maximum of seven crossovers per couple. So, two or three extra bits were enough for that purpose. More generally the last $log_2(max\_cross + 1)$ bits of each individual are used to find an expected optimum number of crossovers. In that way we have two searching spaces: one corresponding to the objective function and other associated to the number of crossovers to apply.

Our attempt is that the individuals preserve the 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 adaptive technique. Once the couple was selected we check 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.* This value is inherited by each children.
- Otherwise we choose a random number in the permitted range and preservation of information is done according to strategy S1 or S2 where,
  - S1, preserves parent's information, enforcing population diversity in the parameter searching space, because most of the time one child inherits characteristics (*ncross_field*) from one of the parent and the other child inherits features from the other parent. (See Fig. 2).
  - S2, preserves individual information (number of crossovers applied when the child was created). This strategy generates more similar individuals (same *ncross_field*) in the parameter searching space and increases loss of genetic diversity. (See Fig. 3).

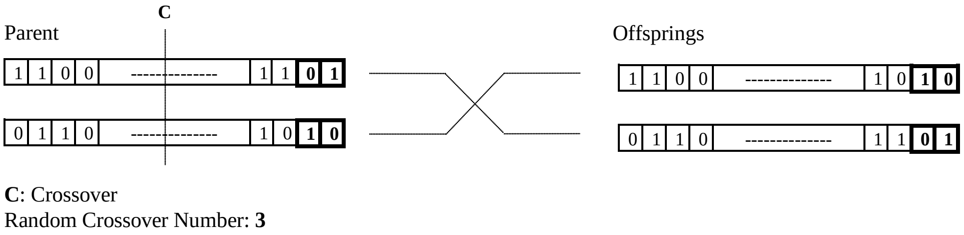


Fig. 2. Strategy S1, three crossover operations applied on parents, children carry parent´s information.

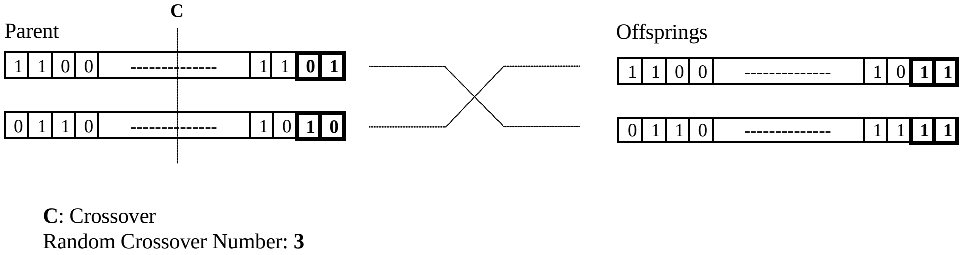


Fig. 3. Strategy S2, three crossover operations applied on parents, children carry their own information.

Experimental test showed that the behaviour 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 vice versa. This behaviour favours the evolutionary process.

2.1.3. BINDING MCPC TO ALTERNATIVE SELECTION MECHANISMS

In this work we studied the effect of MCPC when it was jointly applied to deterministic dynamic ranking selection (DDRS) in order to moderate the combined effect of selection (PS) and MCPC. Baker introduced the first approach to ranking, called linear ranking, in 1985. By means of linear ranking the selective pressure can be controlled by the user. The Baker's original linear ranking method assigns a selection probability that is proportional to the individual's rank. Here, according to Bäck [1] the mapping *rank:* $I \to \{1,...,\mu\}$ is given by:

$$\forall i \in \{1,...,\mu\}: rank(a_i) = i \Leftrightarrow$$
$$\forall j \in \{1,...,\mu - 1\}: f(a_j) \lesseqgtr f(a_{j+1})$$

where ≤ ≥ denotes the ≤ relation or the ≥ relation for minimization or maximization problems respectively. Consequently the index i of an individual a_i denotes its rank. Hence, individuals are sorted according to their fitness resulting a_1 the best individual and a_μ the worst one. Assuming that the expected value for the number of offspring to be allocated to the best individual is $\eta_{max} = \mu P(a_1)$ and that to be allocated to the worst one is $\eta_{min} = \mu P(a_\mu)$ then

$$P_{sel}(a_i) = \frac{1}{\mu}\left(\eta_{max} - (\eta_{max} - \eta_{min}) \cdot \frac{i-1}{\mu - 1} \right)$$

As the following constraints must hold

$$P_{sel}(a_i) \geq 0 \ \ \forall i$$

$$\sum_{i=1}^{\mu} P_{sel}(a_i) = 1$$

it is required that:

$$1 \leq \eta_{max} \leq 2 \text{ and } \eta_{min} = 2 - \eta_{max}$$

The selective pressure can be adjusted by varying η_{max} . As remarked by Baker if η_{max} = 2.0 then the population is driven to convergence during every generation. To restrain selective pressure, Baker recommended a value of η_{max} =1.1. This value for η_{max} close to 1 leads to $P_{sel}(a_i) \simeq 1/\mu$, almost the case of random selection.

It is not an easy task to tune η_{max}, the expected value for the number of offspring for the best individual. This parameter influences selective pressure. Here we propose Deterministic Dynamic Ranking selection (DDRS), a deterministic and dynamic method to update this 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.

2.2. MCMP: THE LATEST MULTIRECOMBINATIVE APPROACH

Attempting to build a better Pareto front in multiobjective optimization, MCMP was born by combining our previous ideas on multiple crossovers and those from J. Lis and A. Eiben [5] in their *multisexual genetic algorithm* (MSGA). This first version of MCMP:

- Uses proportional selection
- Selects multiple parents per sex
- Uses an extension of MCPC (called MCPMA multiple- crossover per mating action).
- For insertion in the next population, it gives preference to those offspring which are classified so far as globally non-dominated.

To build the new population, each time the new offspring are created by application of MCPMA, we apply the following procedure:

```
While the new population is created
  do
     Select $n_1$ parents from each sex,
     Apply MCPMA with uniform scanning crossover to obtain $n_2$ offspring and mutate,
     By consulting $P_{current}$ determine the subset $O_{nond}$ of these new offspring that are globally
     nondominated,
     If $O_{nond} \neq \Phi$ then insert $O_{nond}$ into the new population
                   else insert $n_2/2$ offspring randomly chosen into the new population
  od
```

The number n_1 of parents and the number n_2 of crossovers are parameters of the GA. MCMP was tested on a set of selected multiobjective problems. We show here the results when the new approach is applied to the Problem 3: Schaffer function F2 [15] defined as follows:

Minimize $f_{21}(x)$ and $f_{22}(x)$ where

$$f_{21}(x) = x^2$$

$$f_{22}(x) = (x-2)^2$$

with $-6 \leq x \leq 6$

With the following parameter set,

Population size	:	100
Crossover rate	:	0.85
Mutation rate	:	0.01
Chromosome length	:	14

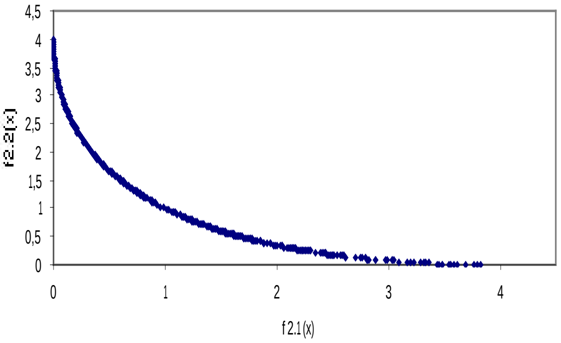


Fig. 4 – The Pareto front for Problem 3, with 3 parents per sex and 3 crossovers.

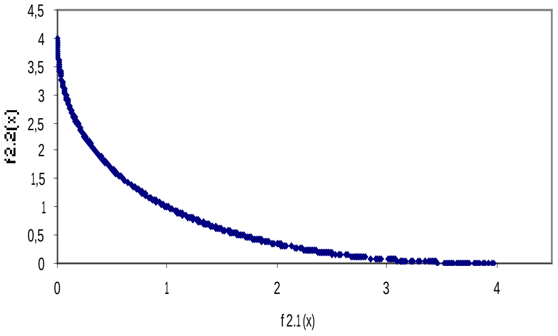


Fig. 5 – The Pareto front for Problem 3, with 4 parents per sex and 4 crossovers.

2.2.1. MCMP FOR SINGLE OBJECTIVE OPTIMIZATION

After the outstanding results obtained in multiobjective optimization, MCMP was tested on single unimodal and multimodal optimization. The following relevant performance variables were examined:

Ebest = ((*opt_val* - best value)/*opt_val*)100

It is the percentile error of the best found individual when compared with the known, or estimated, optimum value *opt_val*. It gives us a measure of how far are we from that *opt_val*.

Epop = ((*opt_val*- pop mean fitness)/*opt_val*)100

It is the percentile error of the population mean fitness when compared with *opt_val*. It tells us how far the mean fitness is from that *opt_val*.

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

All the values analysed were mean values obtained from twenty series completed for each fixed number of crossovers, on each function. Several testing functions were used. We show here results on the Griewank´s and the Easom´s functions when contrasting MCMP and MCPC combined with FCPS.

Performance Variable	Minimum Value		Maximum Value	
	MCPC-FPCS	MCMP	MCPC-FPCS	MCMP
Mean *Ebest*	0.0124	0.0000	2.5176	0.0415
Mean *Epop*	44.9970	0.0260	47.6402	12.3149
Mean *Gbest*	839	45	2340	365

Table 1. Performance variables values for Griewank’s function

Performance Variable	Minimum Value		Maximum Value	
	MCPC-FPCS	MCMP	MCPC-FPCS	MCMP
Mean *Ebest*	0.0381	0.0074	0.0836	0.1634
Mean *Epop*	7.1384	0.0074	9.8447	0.1635
Mean *Gbest*	2026	73	4017	491

Table 2. Performance variables values for Easom´s function

The use of *multiple crossovers on multiple parents* (MCMP) showed to be efficient in optimization of hard unimodal and multimodal testing functions and behaves better than MCPC-FPCS. There is an indication that the multiparent approach mitigates the possible loss of diversity generated by *multiple crossovers per mating* (MCPMA) and no extra adjustments, used before, seem to be necessary. On the other hand, it was shown that the multiparent approach behaves better when it is associated to the multiple crossover approach on both functions selected for optimization. Speed of convergence, measured in number of generations, is augmented without increasing the risk of premature convergence. Consequently the quality of results are better than previous attained under more complex approaches. Additionally, when observing the final population it was detected that all individuals are much more centred surrounding the optimum. This property is strongly detected in the multimodal optimization. This is an important issue when an application requires provision of multiple alternative near-optimal solutions.

3. PARALLEL GENETIC ALGORITHMS

Parallel implementations of Genetic Algorithms (GAs) aim at improvements on performance. In his earlier works Holland [12] recognised the parallel nature of the reproductive paradigm and the intrinsic efficiency of parallel processing. Parallel genetic algorithms (PGAs), models and implementations [13], [16] are designed to exploit this inherent parallel nature of genetic algorithms. When implemented as an island model, on behalf of the evolutionary process, migration of individuals allows for a fruitful interaction between subpopulations by exchanging selected individuals and improving genetic diversity. This exchange is done by choosing an individual from a source subpopulation and exporting it towards a target subpopulation. On arrival, it is usual, for the imported string to be accepted and inserted into the target subpopulation without exerting any control policy. Our earlier experiments [14] controlling migration acceptance showed an improvement of results when contrasted with those obtained by ordinary migration approaches.

In this work we describe extended implementations of alternative strategies to control migration in asynchronous schemes for an *island model*. All of them are an effort to decrease the risk of premature convergence. A first strategy, *Maximum Gap Allowed (MGA),* tries to prevent unbalanced propagation of genotypes by using an acceptance threshold parameter for incoming strings. A second one, *Dynamic Arbiter Strategy (DAS),* permits independent evolution of subpopulations but acts when a possible stagnation is detected. In such condition an attempt to evade falling towards a local optimum is done by inserting an expected dissimilar individual to improve genetic diversity. This is done by exchanging data associated with the best and worst global individuals and population mean fitness. A third alternative, *Combined MGA-DAS Strategy (CMGA-DAS),* combines both of these strategies. The results presented are those obtained in the functions that proved to be more difficult for the *island model* using a simple GA. Experiments were conducted implementing both, virtual and real nodes. The following sections describe the experiments and some results.

3.1. The strategies

MGA, was devised to avoid falling towards a local optimum by introduction of high performers. A parameter θ, was defined as the maximum difference accepted between the fitness of the best local individual and that of the incoming string. Insertion is allowed only when the following condition holds:

$$\text{Fitness}_{\text{ext}} - (1+\theta\ )\,\text{Fitness}_{\text{bestlocal}} \le 0 \quad (\,0 \le \theta \le 1)$$

This strategy was applied with an interconnectivity scheme of a static logical ring; if the number of processors is n then $\text{node}_{(i+1)\ \text{mod}\ n}$ is the neighbour of node_i .

DAS, decides by means of a global arbiter if a migrated chromosome should be inserted or not into some subpopulation. This decision is based on the knowledge the arbiter has about the evolutionary progress of subpopulations, hence exerting a sort of dynamic convergence control. At migration time, rather than sending a single chromosome, the process managing the chromosome exchange exports a packet to the arbiter containing data about; source node address, best individual chromosome, worst individual chromosome, best individual fitness, worst individual fitness, and subpopulation mean fitness. On its end, at each migration arrival, the arbiter updates information about the best and worst global individuals and subpopulation fitness. Also, information about the best individual of the first migration is kept on hand. In more detail, when the arbiter receives a packet, from the source, the following actions take place:

- If it is the first migration, then updates its internal data structures.
- Otherwise, updates its internal data structures and to determine the progress of the evolutive process, compares the current mean fitness value of the source subpopulation with the last updated corresponding value and,
  - ◊ If they remain similar (possible search stagnation) a migration of an individual to the source subpopulation will take place.
  - ◊ Otherwise (search improves results) no action take place.

To determine which individual to migrate the following criterion was adopted:

```
*if the best global individual does not reside in the source subpopulation*
    *then migrate the best global individual*
    *else migrate the worst global individual.*
```

Giving the arbiter the faculty to migrate (or not) a global individual (originated in any node) to the source node, resulted in a *dynamic interconnection scheme*.

Finally, *Combined MGA-DA Strategy* (CMGA-DAS), consisting of the combined application of both previous strategies, was also examined by simply adding to DAS the acceptance criteria imposed by θ, when determining which individual to migrate. So, the migration criterion applied for this strategy was:

```
if the best global individual resides in the source subpopulation
  then migrate the worst global individual
  else if $\theta$ test holds for the  best global individual
         then migrate the best global individual
         else if $\theta$ test holds for the  best first migrated individual
                then migrate the best first migrated individual[2]
                else migrate the worst global individual
```

3.1.1. Experiments and Results

A set of, at least, twenty runs was performed for our experiments. The *island model* was run on the set of several test functions, solving optimization problems. Only the results on the *f2* Volcano (hard unimodal) and the *f4* Schaffer F7 (hard multimodal) functions are referred here. A simple GA for each subpopulation was used, applying: proportional selection (for mating), tournament selection (for replacement), elitism, one-point crossover and bit-swap mutation, on a population of 70 individuals. Four parameter sets, S1 to S4, with typical values for probabilities of crossover and mutation were used. The number of generations was limited to 4000. To achieve subpopulation interaction, with and without migration arbitration, sets of 6, 10 and 16 nodes were used. After the runs were completed, mean values for *Ebest* and optimal hits (as below defined) were determined:

Optimal Hits = (# optimal hits / # runs). The hit ratio to find the optimal solution, throughout the total number of runs.

The following tables and graphs show a report of experimental results. All the values in the tables are mean values obtained from the multiple run series.

In figures 6 and 7 we observe that those strategies based on the acceptance threshold parameter θ, are the best performing ones with this very hard deceptive unimodal function.

In figure 8 we observe that although the *Ebest* values are quite small, none of the contrasted strategies reached the optimum frequently. Optimal hits obtained in the best case was of 28%, with parameter set S_1 and 16 nodes. PGA implementations are notably superior than sequential GA implementation. No simple sequential GA can even approach the worst near optimal solution found by any strategy in similar tests.

[2] The best first migrated individual is a good intermediate value which contributes to genetic diversity.

# *nodes*	*Static*	*MGAS*	*DAS*	*CMGA-* *DAS*
6	7.61E-03	6.92E-04	1.66E-02	2.08E-03
10	2.08E-03	0.0	6.23E-03	0.0
16	6.92E-04	0.0	1.04E-03	0.0

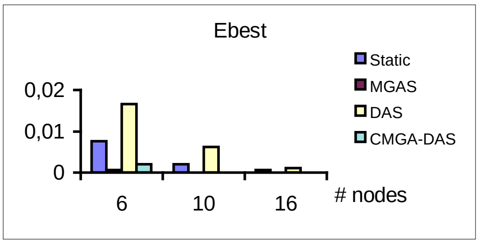


Fig 6. Ebest values for f2 function
under each strategy with set S2
for variable number of nodes

# *nodes*	*Static*	*MGAS*	*DAS*	*CMGA-* *DAS*
6	0.81	0.95	0.75	0.95
10	0.95	1.0	0.916	1.0
16	0.983	1.0	0.95	1.0

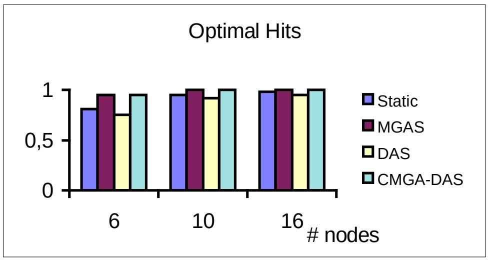


Fig 7. Optimal hits for f2 function
under each strategy with set S2
for variable number of nodes

# *nodes*	*Static*	*MGAS*	*DAS*	*CMGA-* *DAS*
6 (S1)	7.29E-13	1.28E-12	7.48E-13	7.67E-13
6 (S2)	6.36E-13	1.62E-12	1.47E-12	1.57E-12
6 (S3)	4.41E-12	4.28E-12	4.67E-12	1.54E-11
6 (S4)	2.14E-12	1.52E-12	1.92E-12	4.50E-12
10 (S1)	2.00E-13	4.32E-13	6.81E-13	3.34E-13
10 (S2)	3.83E-13	3.11E-13	5.60E-13	9.22E-13
10 (S3)	1.04E-12	1.54E-12	8.41E-13	9.82E-13
10 (S4)	5.51E-13	8.75E-13	1.22E-12	8.00E-13
16 (S1)	1.17E-13	1.73E-13	3.62E-13	2.56E-13
16 (S2)	1.41E-13	2.34E-13	5.62E-13	4.54E-13
16 (S3)	5.36E-13	4.82E-13	1.41E-12	1.01E-12
16 (S4)	3.67E-13	3.50E-13	1.31E-12	7.77E-13

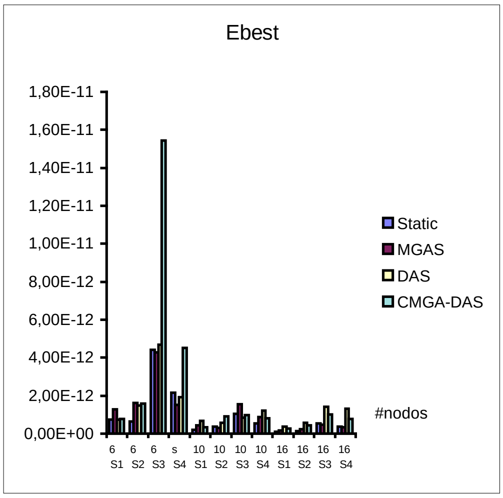


Fig 8. Ebest values for f4 function under each strategy over all parameter sets for variable number of nodes.

4. CONCLUSIONS

This paper shows two major contributions to the theoretical field of Evolutionary Computation. Topics involving different multirecombination schemes, their effect when applied to single and multiple criteria optimization problems and parameters adaptability, were investigated carefully. Also a set of strategies to control migration in parallel genetic algorithms were considered.

Multiple crossovers per couple (MCPC) showed its benefits and limitations, described in detail in previous sections. To overcome these limitations successful approaches were undertaken by combining MCPC with FPCS, by using self-adaptation of MCPC parameters or by binding MCPC to alternative selection mechanisms. The use of *multiple crossovers on multiple parents* (MCMP) proved to be efficient in single and multiple objective optimization and behaves better than previous improvements. Speed of convergence, measured in number of generations, is augmented without increasing the risk of premature convergence. Consequently the quality of results is better than those previous attained under more complex approaches. There is indication that the multiparent approach mitigates the possible loss of diversity and no extra adjustments seem to be necessary. Additionally, by observing the final population it was found that all individuals are much more centred surrounding the optimum on both function optimizations and this is even more so in the multimodal optimization. This property was not observed neither with other previous approaches nor with the multiparent original approach. This is an important issue when an application requires provision of multiple alternative near optimal solutions. On the other hand, it was shown that the multiparent approach behaves better in accuracy of results and speed when it is associated to the multiple crossovers approach on both functions selected for optimization. Although we cannot be conclusive, we conjecture that by means of this association the searching space is efficiently exploited by the multiple application of crossovers and efficiently explored by a greater number of samples provided by the multiple parents. In view of these promising results new work is currently being developed to study the optimal (n_1,n_2) association, the consequences of increasing the number of crossovers, and the effect of multiple crossovers on multiple parents under diverse crossover methods.

Three new strategies to control migration in asynchronous Parallel Genetic Algorithms distributed in a network of 6, 10 and 16 processors have been discussed. Here, it is worth remarking that the base for the evolutionary approach, upon which results were completed, is the weakest one; a simple GA. Two kinds of problems were addressed for optimisation: unimodal and multimodal. Easom's and the (hardest) Volcano functions are good representatives of the first class of problems; to find a needle in a haystack. For them, MGAS and CMGA-DAS were the strategies showing better performance. In every case *Optimal Hits* increases accordingly with increments in the number of processors, arriving at 100% under MGAS and CMGA-DAS for 10 and more nodes. For the second class of problem, difficult highly multimodal functions of varied landscapes were chosen. Here there cannot be detected a clear preeminence of one strategy over the others and for any parameters set Static, MGAS and CMGA-DAS work better. Further studies are needed to ensure the utility of the new proposed strategies for these types of functions. Fine tuning of genetic operators probabilities and knowledge of the degree of population convergence are prospective issues to investigate. We want to remark that PGA implementations are notably superior than sequential GA implementations in view of quality of results. No simple sequential GA can even approach the worst near optimal solution found by any strategy in similar tests. Finally in the research field of Evolutionary Computation future work is addressed to combine the new multirecombinative approaches and their parallel implementations.

5. Acknowledgements

We acknowledge the cooperation of the project group for providing new ideas and constructive criticisms. Also to the Universidad Nacional de San Luis and the ANPCYT from which we receive continuous support.

6. Bibliography

[1]. Bäck T: *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.

[2]. Belding T.C. - *The Distributed Genetic Algorithm Revisited* - Proceedings of the Sixth International Conference on Genetic Algorithms, pp 114-121, Morgan Kauffman, 1995.

[3]. Cohoon J.P., Hedge S.U., Martin W.N., Richards D. - *Puntuacted Equilibria: A Parallel Genetic Algorithm* - Proceedings of the Second International Conference on Genetic Algorithms, pp 148-154, Lawrence Erlbaum Associates Publishers, 1987.

[4]. Easom, E.: *A Survey of Global Optimization techniques*. M. Eng. Thesis, Univ. Louisville, Louisville, KY, 1990.

[5]. Eiben A., Lis J., *A multisexual genetic algorithm for multiobjective optimization*, 4th IEEE International Conf. on Evolutionary Computation (ICEC'97), pp 59-64, Indianapolis, USA, April 1997.

[6]. Esquivel S., Gallard R., Michalewicz Z., - *MCPC: Another Approach to Crossover in Genetic Algorithms*- Proceedings of the 1st Congreso Argentino de Cs. de la Computación, pp 141-150, Universidad Nacional del Sur, October 1995.

[7]. Esquivel S., Leiva A., Gallard R., - *Multiple crossover per couple in genetic algorithms*. Proc. of the 4th IEEE International Conf. on Evolutionary Computation (ICEC'97), pp 103-106, Indianapolis, USA, April 1997.

[8]. 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), Vol. 1, pp 235-241, La Laguna, Tenerife, Spain, February 1998.

[9]. 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), pp 419-425. Universidad Nacional del Comahue, Argentina, October, 1998.

[10]. 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), pp 383-391. Universidad Nacional del Comahue, Argentina, October 1998.

[11]. Esquivel S., Leiva H.,.Gallard R., *Multiple Crossovers Between Multiple Parents To Improve Search In Evolutionary Algorithms.* Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 1589-1594.

[12]. Holland J. – *Adaptation in Natural and Artificial Systems* – Ann Arbor, MI: University of Michigan Press. 1975

[13]. Levine, D. - *A Parallel Genetic Algorithm for the Set Partitioning Problem* - Ph D Thesis, Illinois Institute of Technology and Argone National Lab. (ANL -94/23), 1994.

[14]. Ochoa C., Gallard R. - *Strategies for Migration Overseeing in Asynchronous Schemes of Parallel Genetic Algorithms* - Proceedings of the International ICSC Symposium on Soft Computing: SOCO'97.

[15]. Schaffer J. D., *Some experiments in machine learning using vector evaluated genetic algorithms*, Doctoral dissertation, Department of Electrical Engineering, Vanderbilt University. 1984.

[16]. Shyh-Chang Lin, Punch W., Goodman E. - *Coarse-Grain Parallel Genetic Algorithms: Categorizations and New Approach*. Parallel & Distributed Processing, Dallas TX, Oct. 1994.

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

[18]. Tanese R. - *Distributed Genetic Algorithms* - Proceedings of the Third International Conference on Genetic Algorithms, pp 434-439, Morgan Kauffman, 1989.

[19]. Whitley D. - *An Executable Model of a Simple Genetic Algorithm* - Fundations of Genetic Algorithms-2 (FOGA-92), pp 45-62, Morgan Kaufmann, 1993.