UPGRADING EVOLUTIONARY ALGORITHMS THROUGH MULTIPLICITY FOR MULTIOBJECTIVE OPTIMIZATION IN JOB SHOP SCHEDULING PROBLEMS

ESQUIVEL S. C., FERRERO S.W., GALLARD R.H.

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: {esquivel, swf, rgallard}@unsl.edu.ar
Phone: + 54 652 20823
Fax : +54 652 30224

Abstract

In previous works the ability of CPS-MCPC (an evolutionary, co-operative, population search method with multiple crossovers per couple) to build well delineated Pareto fronts in diverse multiobjective optimization problems (MOOPs) was demonstrated. To test the potential of the novel method when dealing with the Job Shop Scheduling Problem (JSSP), regular and non-regular objectives functions were chosen. They were the makespan and the mean absolute deviation (of job completion times from a common due date, an earliness/tardiness related problem). Diverse representations such as priority list representation (PLR), job-based representation (JBR) and operation-based representation (OBR) among others were implemented and tested. The latter showed to be the best one. As a good parameter setting can enhance the behaviour of an evolutionary algorithm distinct parameters combinations were implemented and their influence studied. Multiple crossovers on multiple parents (MCMP), a powerful multirecombination method showed some enhancement in single objective optimization when compared with MCPC.

This paper shows the influence of different recombination schemes when building the Pareto front under OBR and using the best parameter settings determined in previous works on a set of demonstrative Lawrence´s instances. Details of implementation and results are discussed.

Keywords: Evolutionary Computation, Job shop scheduling, multiobjective optimization, multirecombination.

[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 a MOOP, a solution has a number of objective values, one per each optimizing criterion (attributes). As many of these criteria can be in conflict it is impossible to optimize any of the objective functions without degrading some of the remaining criteria. This leads to a decision-making problem for choosing a suitable solution (or set of solutions) according to higher-level organization goals [23].

Vilfredo Pareto [25] established that there exists a partial ordering in the searching space of a MOOP based on a *domination* relationship. For instance, in a maximization problem given two solutions $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$, the Pareto criterion says that, x dominates y iff $x_i \geq y_i \ \forall i$ and $\exists \, j$ such that $x_j > y_j$.

In the problem space some solutions will not be dominated by any other solution and they conform the *Pareto front*, also known as the *acceptable set*, the *efficient points* and the *Pareto optimal set*. Knowledge of the Pareto front is of utmost importance when search is applied before decision making. This information provides to the judgement of a human decision-maker with the trade-offs to establish interactions among different criteria, hence simplifying the decision process to choose an acceptable range of solutions for a multicriteria problem. Implemented first by Schaffer [26], [27], Fourman [16] and then by Kursawe, [20], [21] and others, *cooperative population searches (CPS) with criterion selection* [19] was used to build the Pareto front in selected multicriteria problems. The central idea in CPS, is to make a parallel single criterion search, where all members of the population of an evolutionary algorithm are involved in a cooperative search to build the Pareto front.

Complexity of scheduling problems [17] and their economical impact motivated extensive research [1], [2], [28], [29]. The job shop scheduling problem (JSSP) is related to the allocation of limited resources (machines) to jobs over time. This is a decision making process that has as a goal the optimization of one or more objectives. The model considered here assumes that the system consists of a number of different machines and only one job may execute on a machine at a time. All schedules and jobs are non-preemptive. Jobs can have distinct priorities and all of them are available at production initiating time. Each job visits all machines, only once, following a predetermined sequence of machines, called a *route*.

Due to their implicit parallel search, evolutionary algorithms (EAs) are suitably fitted to deal with JSSP [7], [14], [24] as well as seeking solutions in multiobjective optimization [3], [4], [5], [6], [10], [15].

The present work investigates the ability of the CPS-MCMP method, a co-operative population search approach allowing multiple crossovers applied on multiple (2 or more) parents, to find non-dominated points and contrasts its performance against other recombination schemes when building the Pareto front.

2 SELECTED OBJECTIVES, REPRESENTATION AND OPERATORS FOR THE JSSP-MOOP

For multiobjective optimization of a JSSP we selected f_1 as the *makespan*, and f_2 as the *mean absolute deviation* (MAD) of job completion times from a common due date d, as the conflicting criteria to minimize. When minimizing function f_1, schedules tend to be shortened, usually implying high utilization of machines. When minimizing function f_2 earliness and tardiness are penalized at the same rate for all jobs and schedules are built so that d is in the middle of the job completion times, which usually

derives in lower inventory costs. Consequently, for the JSSP and given a due date *d*, common to all jobs, our multiobjective optimization problem can be formulated as follows:
Minimize $f_1(\sigma)$ and $f_2(\sigma)$ where sought solutions σ are feasible schedules and

$$f_1(\sigma) = max_{1 \leq k \leq m}\{max_{1 \leq i \leq n}\{C_{ik}\}\} \quad (1) \qquad\qquad f_2(\sigma) = \frac{1}{n}\sum_{j=1}^{n}\left|C_j - d\right| \qquad (2)$$

In expression (1) C_{ik} stands for the completion time of the last operation of job *i* in machine *k*, and in expression (2) C_j indicates the completion time of the last operation of job *j*.

Regarding representation, a particular encoding of solutions imposes limitations on the genetic operators to be used. This issue is mainly addressed by the creation of valid offspring avoiding the use of penalties or repair algorithms. In our previous experiments operation-based representation provided best results. Consequently, we adopted this encoding technique for the work described here. Under OBR a schedule is encoded in the chromosome as a sequence of operations. Due to the existence of precedence constraints among operations of a particular job, the assignment of natural numbers to identify operations and the use of a permutation representation can lead to infeasible schedules. To avoid this problem Gen, Tsujimura and Kubota [18] proposed a representation where each operation is identified by the job number to whom it belongs and the order of occurrence in the sequence. For an *n*-job *m*-machine problem, a chromosome consists of *n* x *m* genes, where each gene has a job identifier as the allele value and values are repeated exactly *m* times in the chromosome.

Regarding operators, our proposal is *modified order cross*over (MOX). To build a valid offspring a sub-sequence of one parent is inserted in the same position in the offspring and the rest of allele values are copied from the second parent in the order they are appearing controlling the number of allele repetitions. For mutation a *modified exchange mutation* (MXM) was implemented in order to ensure that the exchange effectively changes the allele values. Detailed information and examples on OBR and MOX can be seen in [8]

3 MULTIRECOMBINING COOPERATIVE POPULATION SEARCHES (CPS-MCMP)

Independently of the method being used, conventional approaches to crossover involve applying the operator only once on the selected pair of parents. Such a procedure is known as the Single Crossover Per Couple approach (SCPC). In earlier works [12] [13], we devised a different approach to crossover which allows multiple offspring per couple (MCPC) , as often it happens in nature. To deeply explore the recombination possibilities of previously found solutions, 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 , Schaffer F_6 and other functions). A simple genetic algorithm, with conventional operators and parameter values, was the basis of those initial experiments. By combining the Eiben´s single crossover multiparent (SCMP) approach and MCPC further studies gave raise to the extension known as multiple crossovers on multiple parents (MCMP). The latest member of the multirecombinative family showed its advantages in many single and multiobjective problems [10] [11]. For multiobjective optimization initial experiments with CPS-MCPC were implemented executing exactly n_1 crossovers, providing $2\ n_1$ children per couple ($2 \leq n_1 \leq 4$). Implementation details can be seen in [8]. The method presented here CPS-MCMP applies n_1 crossovers on n_2 parents. Basically this approach:

1) Maintains a single population of solutions that is separately ranked according to each criterion.
2) Uses ranking selection to select $n_2/2$ parents per criterion.
3) Uses *multiple crossovers multiple parents* (MCMP), and the corresponding crossover and mutation operators to generate multiple offspring.
4) After each mating n_3 individuals are selected for insertion in the next population. The algorithm first selects those offspring, which are classified so far, as *globally* non-dominated. If none fulfilling this condition exists then half of the m newly generated offspring are inserted, selecting first those that are non-dominated within the new offspring subset and completing $m/2$ insertions by random selection if necessary. So n_3 is equal to the number of globally non-dominated offspring, or it is equal to $m/2$.

The value of m stands for the size of the offspring subset and its value is given by:

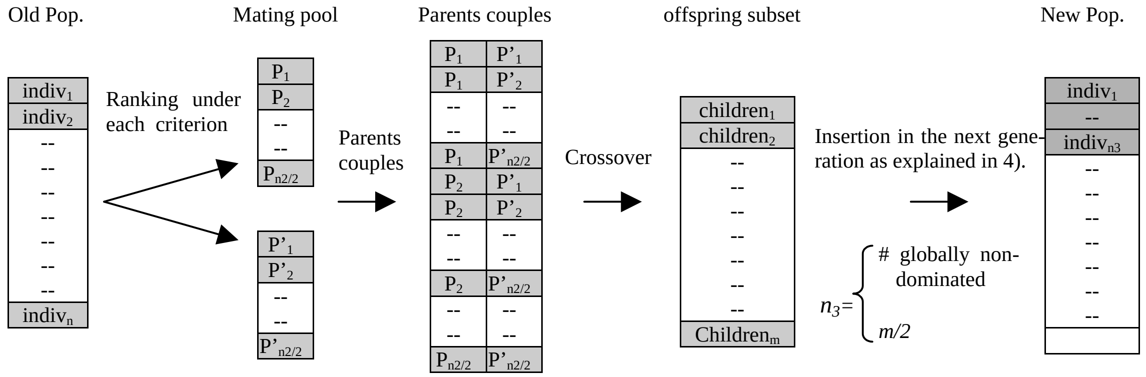
n_1: number of crossovers.
n_2: number of parents.

$$m = \left({n_2}/{2} \right)^2 * n_1 * 2$$

Point 4 above mentioned, implies to maintain the updated set of solutions found so far as belonging to the Pareto front, which is called $P_{current}$.

As we previously said a particular representation imposes limitations on operators. Under OBR an alternative multi-parents and multi-crossovers recombination was necessary to be considered:

1. The same number ($n_2/2$) of parents is selected under each criterion to conform the mating pool.
2. Each parent selected under the first criterion mates (only) every other parent selected under the second criterion. Recombination at this stage applies SCPC or MCPC on each couple of parents.



The values of n_1 and n_2 selected for the work reported here (table 1) were chosen to study the influence of different recombination approaches in a population limited to 100 individuals, as in previous experiments.

n_1	n_2	Recombination method	Offspring subset size (m)
1	**2**	SCPC	2
3	**2**	MCPC	6
1	**4**	SCMP	8
2	**4**	MCMP	16
1	**6**	SCMP	18
2	**6**	MCMP	36

Table. 1. Set of considered experiments for different values of n_1 and n_2.

4 EXPERIMENTS

The problem of minimizing $f_1(\sigma)$ and $f_2(\sigma)$, was used to evaluate the performance of the CPS-MCMP and CPS-MCPC approaches. Three instances of two types, small and medium (size) from the Lawrence's benchmark set [22], with known optimal makespan values were used. Small instances of 10 jobs and 5 machines, are identified as *la01* and *la02*, while the medium instance of 20 jobs and 10 machines is identified as *la30*.

Best parameter setting was determined by taking previously found results for this problem presented elsewhere [9]. For the experiments discussed here parameters were set as follows. Crossover and mutation probabilities fixed at 0.65 and 0.005, respectively. One of the main conclusions from the previous work is that the algorithm keeps evolving still in advanced generations, so a maximum number of generations was fixed at 50000. The population size was fixed at 100 individuals. Elitism was used to retain the best individual found so far under each criterion. As optimal values of the makespan were known for each instance of the test suite, the common due date *d* to determine $f_2(\sigma)$ values was fixed at a value 40% greater than the corresponding optimal makespan.

To compare the diverse algorithms two performance measures were proposed:

- *Pareto Front Quality (PFQ)*: A Pareto frontier A is said to have higher quality than another Pareto frontier B if most of the points in A dominate the points in B.
- *Pareto Front Size (PFS)*: indicates the total number of globally non-dominated points created during the evolutionary process as the number of generations is increased.

5 RESULTS

In the following graphics the captions *i*C-*j*P-X stands for n_1 crossovers and n_2 parents and X efficient points in the frontier. For the discussion of the compared performance of multirecombined methods on small instances we will show only results for *la01* as demonstrative instance, because *la02* reveals similar findings

Figure 2 shows that as the number of parents is augmented, maintaining fixed at 1 the number of crossovers, the quality of the frontier (PFQ) is improved and the size of the frontier (PFS) becomes larger.

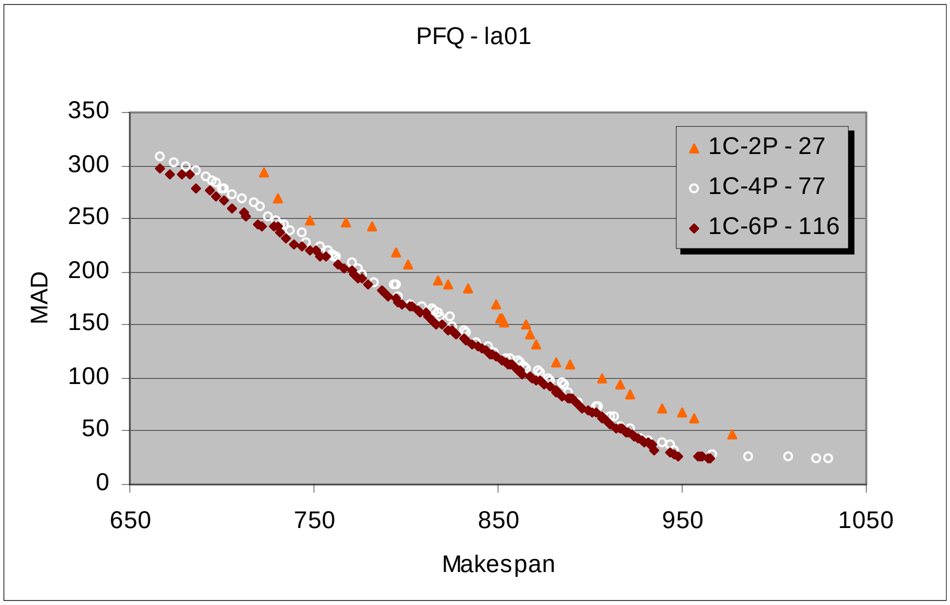
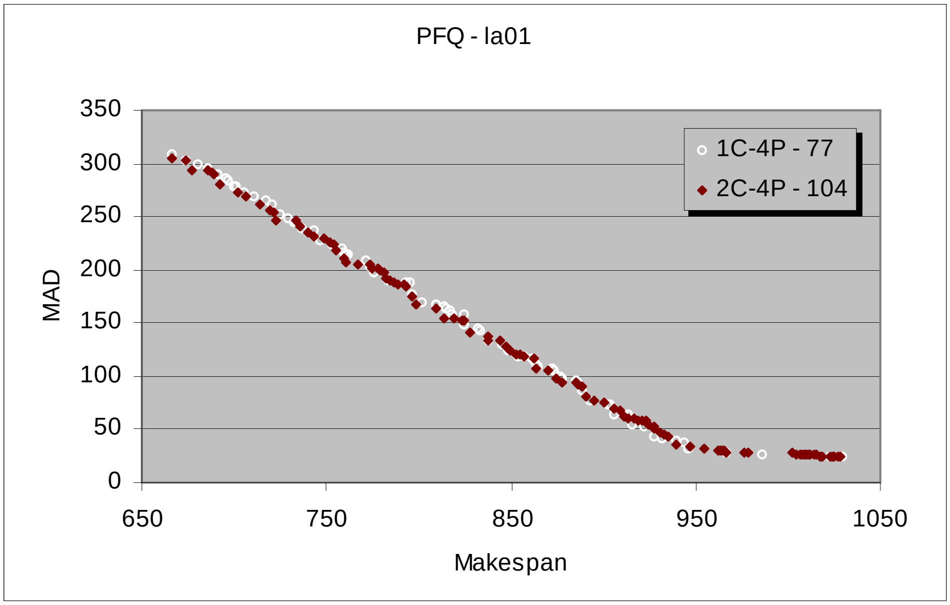


Figure 3 shows that as the number of crossovers is augmented, maintaining fixed at 4 the number of parents, PFQ is slightly improved while PFS becomes considerably larger and denser. Similar results were observed when contrasting 1C-6P versus 2C-6P.



Under MCMP best, and similar, results in *la01* were obtained with n_1 = 1 and 2 and n_2 = 6. In previous works best results under MCPC were obtained with n_1=3. Figure 4 shows better quality frontier and larger frontier size under greater parents multiplicity.

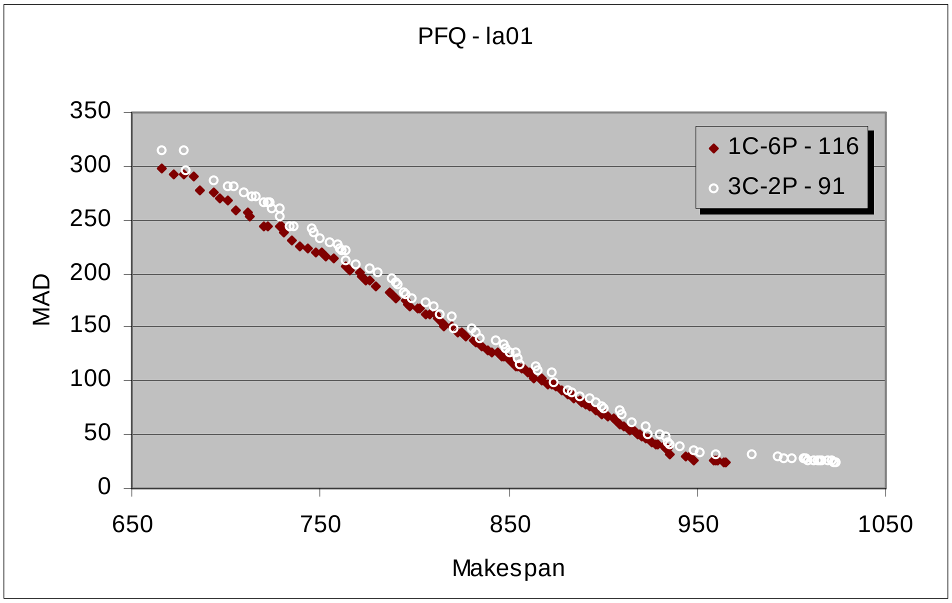
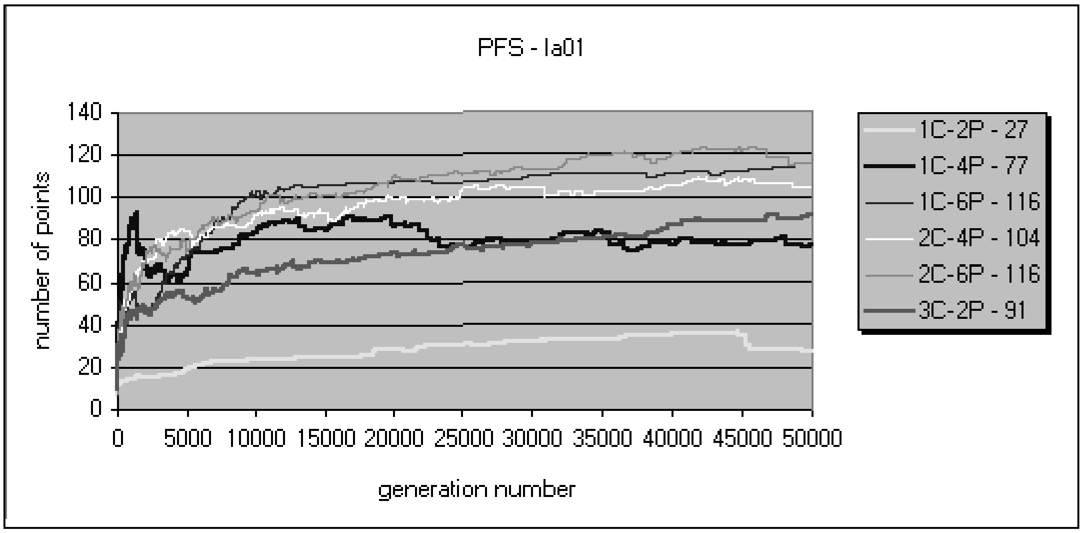


Figure 5 shows how the efficient points are accumulated through the generations. It is observed here that PFS, after 50000 generations, is greater and denser as long as n_1 and n_2 allow better exploration and exploitation. Best results, reaching 116 points, are obtained with 1C-6P and 2C-6P. The number of parents has a stronger influence than the number of crossovers in PFS.



Regarding medium size instance *la30* best quality partial frontier was obtained under 2C-4P for MS values greater than 1656 and MAD values lesser than 337. Under 2C-6P a slightly lower quality frontier is obtained but covering a wider range of values for MS and MAD. This latter frontier is contrasted in figure 7 with the best quality frontier obtained in a previous work under MCPC with n_1=3.

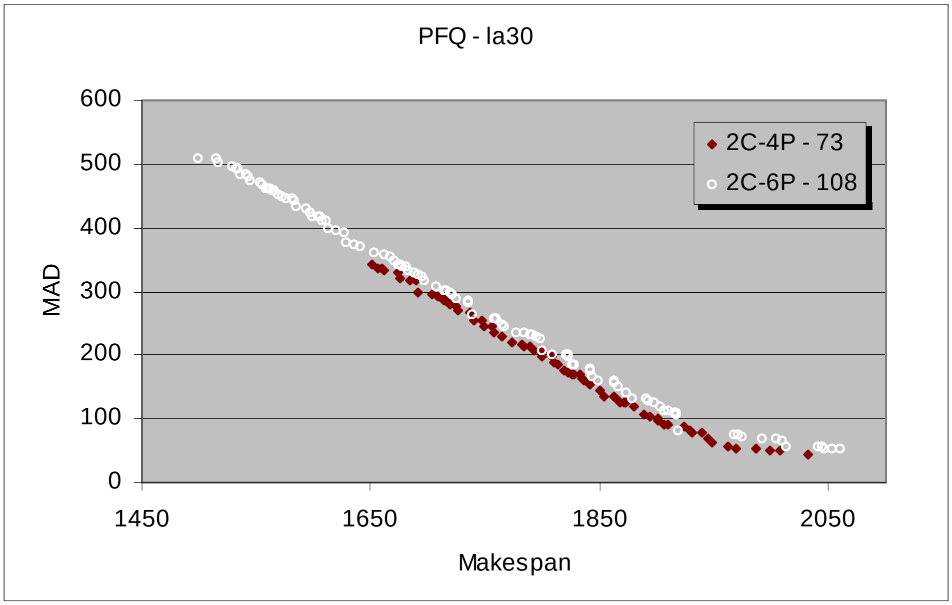


Figure 7 shows that under MCMP (2C-6P) results are better than under MCPC (3C-2P). As it happened for instance *la01*, results are better for PFQ and PFS performance measures.

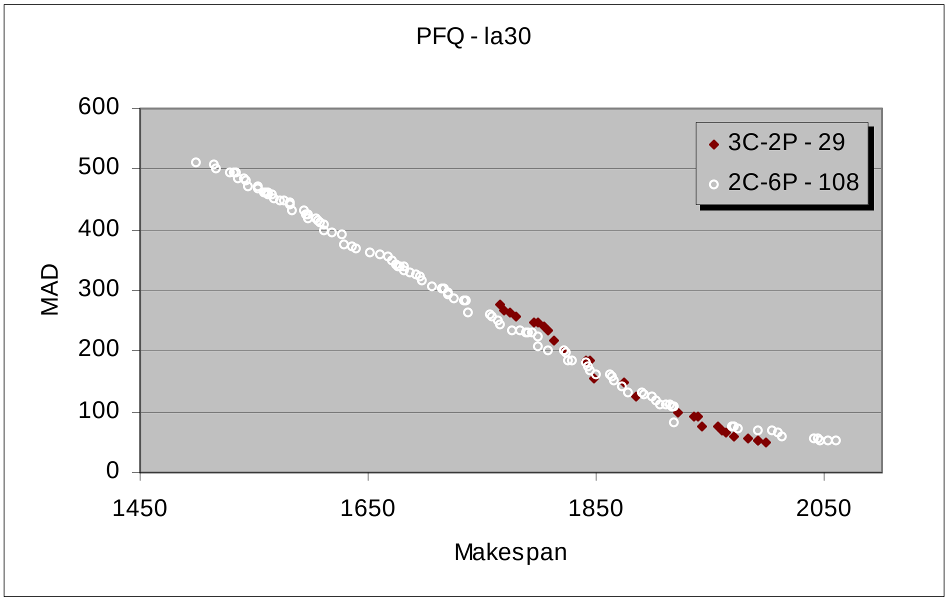


Figure 8 shows that as long as n_1 and n_2 are incremented the algorithm continues accumulating points through the generations. Best results corresponds to MCMP (2C-6P).

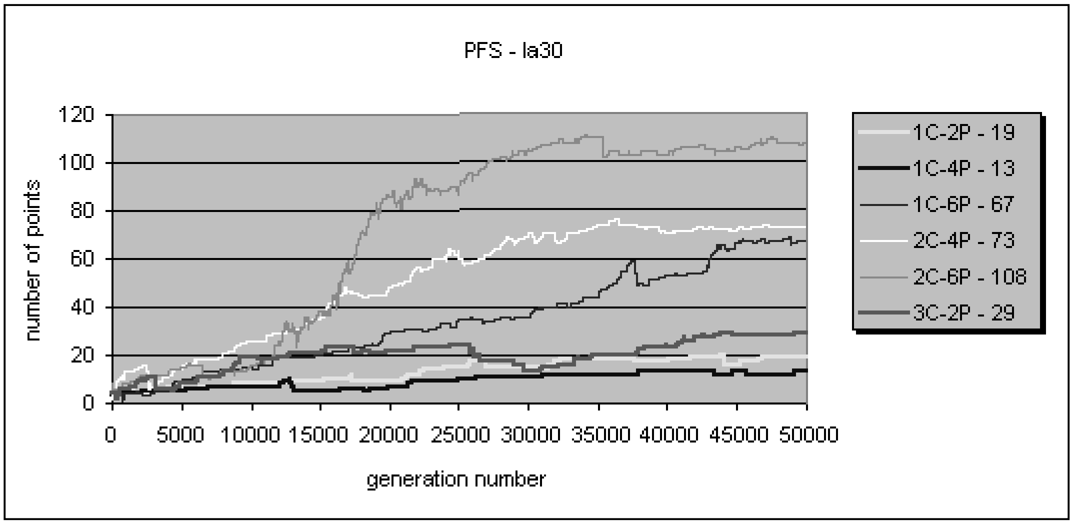


Table 2, shows average PFS values for all considered instances under different recombination approaches. Here we can see that a denser Pareto frontier is built as long as the multiplicity of crossovers and parents is incremented. This benefit is obtained at a cost of approximately a double of computational effort for the multiplicity approaches when compared against the simpler SCPC recombination method.

n_1	n_2	Recomb.	Mean PFS
1	**2**	SCPC	23,67
3	**2**	MCPC	62,00
1	**4**	SCMP	56,67
2	**4**	MCMP	82,33
1	**6**	SCMP	92,00
2	**6**	MCMP	97,67

Table. 2. Mean_PFS results for the 3 instances under different values of n_1 and n_2.

6 CONCLUSIONS

In this work we show different evolutionary approaches to face multiobjective optimization in a set of three selected instances of the Job Shop Scheduling problem. A co-operative population searches method was implemented and distinct recombination approaches were applied and contrasted: SCPC, SCMP, MCPC and MCMP. In all cases OBR, the best representation found in previous works for the JSSP, was used.

To study the behaviour of the algorithms under these different recombination schemes, two performance measures were defined: the quality of the Pareto front obtained (PFQ) and the total number of globally non-dominated points created during the evolutionary process as the maximum number of generations is increased (PFS). This preliminary set of experiments gives the following indications:

As long as the number of parents is incremented, for a fixed number of crossovers, Pareto frontiers of better quality and higher density are obtained. On the other hand as the number of crossovers is incremented, for a fixed number of parents, highest density frontiers with a slight better quality are obtained. This latter effect is clearly detected in medium size instance la30. This indicates that exploitation of good solutions through repetitive application of crossovers provides more non-dominated solutions in larger search spaces.

When compared with conventional SCPC the multiplicity feature added to these evolutionary algorithms provides notably better results regarding quality (PFQ) and size (PFS) of the Pareto fronts discovered through the search. Improvements under MCPC and MCMP are obtained by paying an average cost of about a double of the computational effort required for SCPC. So, when PFQ and PFS are the main objective of the decision-maker the multiplicity feature becomes an advisable alternative.

Future work will include experiments with larger population size, to allow higher degree of multirecombination and the study of parallel implementations to reduce computational time.

7 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.

8 BIBLIOGRAPHY

[1] Baker K., Scudder G., *Sequencing with earliness and tardiness penalties: A review*. Operations Research, vol. 38, pp 22-36, 1990.

[2] Bruns R., *Direct chromosome representation and advanced genetic operators for production scheduling*. In Forrest ICGA93 , pages 352-359.

[3] Coello Coello C., *An updated survey of evolutionary multiobjective optimization*, Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 3-13.

[4] Chen R., Gen M., Tsujimura Y., *A tutorial survey of job shop scheduling problems using genetic algorithms: part I, representation*, International Journal of Computers and Industrial Engineering, vol. 30, no. 4, pp. 983-997, 1996.

[5] Deb K., *Multi-objective genetic algorithms: problem difficulties and construction of test problems*. Evolutionary Computation, 1999 by the MIT.

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

[7] Eiben A., Raue P.-E., Ruttkay Zs., *Genetic Algorithm with multiparent recombination*, Proc of Parallel Solving from Nature Nr 3, LNCS 866, Springer Verlag, pp 78-87, 1994.

[8] Esquivel S., Ferrero S.,.Gallard R. *Multirecombinated evolutionary algorithms to solve multiobjective job shop scheduling*. Proceeding of Primer Congreso Argentino de Ciencias de la Computación, 2000.

[9] Esquivel S. C., Ferrero S.W., Gallard R.H., *Parameter setting in evolutionary algorithms for multiobjective jssp optimization*. Proceedings of the International Conference in Soft Computing. University of Paisley, Scotland, U.K., June 2001.

[10] Esquivel S., Leiva H.,.Gallard R., *Multiplicity in genetic algorithms to face multicriteria optimization*, Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 85-90.

[11] Esquivel S., Leiva H.,.Gallard R., *Multiple crossovers between multiple parents to improve search in evolutionary algorithms*, Evolutionary Computation , IEEE Publishing Co, Washington DC, pp 1589-1594., 1999.

[12] 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.

[13] Esquivel S., Gallard R., Michalewicz Z., *MCPC: Another Approach to Crossover in Genetic Algorithms*, Proceeding of Primer Congreso Argentino de Ciencias de la Computación, pp 141 - 150, 1995.

[14] Fang H.-L., Ross P., and D. Corne. *A promising genetic algorithm approach to job-shop scheduling, rescheduling, and open-shop scheduling problems*. ICGA93 , pages 375-382. Proceedings of the Fifth International Conference on Genetic Algorithms . Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[15] Fonseca C. M., Fleming P. J. *Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization* Proc. of the 5th In. Conf. on Genetic Algorithms, pp 416-423, Urbana- Champaign, IL, Morgan Kaufmann, 1993.

[16] Fourman M. P., *Compaction of symbolic layout using genetic algorithm*, Proc. of the 1st Int. Conf. on Genetic Algorithms, ed Grefenstette J.J pp 141-153, Pittsburgh, PA, 1985.

[17] Garey M. R. and D. S. Johnson. *Computers and Intractability . A Guide to the Theory of NP-Completeness*. Freemann & Co., San Francisco, CA, 1979.

[18] Gen M., Tsujimura Y., Kubota E., *Solving job-shop scheduling problem using genetic algorithms*, Proceedings of the 16th International Conference on Computers and Industrial Engineering, Ashikaga, Japan, 1994.

[19] Horn.J., *Multicriterion decision making*, Handbook of Evolutionary Computation. F1.9:1-9:15, Oxford University Press, 1997.]

[20] Kursawe F., *Evolutionststrategien für die vectoroptimierung*, diplomarbeit, universitat Dortmund, 1990

[21] Kursawe F., *A variant of evolution strategies for vector optimization*, Parallel Problem solving from Nature, Lectune notes in Computer Science 496 Berlin:Springer, pp 193-197. 1991.

[22] Lawrence S., *Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques*, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1984.

[23] Leitmann G., Marzollo A., *Multicriteria Decision Making*, CISM No 211, Springer Verlag, Wien , NY, 1975.

[24] Nakano R. and Yamada T., *Conventional genetic algorithms for job shop problems*. Proceedings of the Fourth International Conference on Genetic Algorithms , pages 474-479. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[25] Pareto V., *Cours d'Economie Politique*, 1896, Switzerland, Lausanne: Rouge.

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

[27] Schaffer J. D., *Multiple objective optimization with vector evaluated genetic algorithms*, Proc. of the 1st Int. Conf. on Genetic Algorithms, ed Grefenstette J.J pp 93-100, Pittsburgh, PA, 1985.

[28] Shaw K. J., Northcliffe A. L., Thompson M., Love J., Fleming P.J., Fonseca C. M, *Assessing thge performance of multiobjective Genetic Algorithms for Optimization of a batch process scheduling problem*, Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 37-45.

[29] Syswerda G. *Schedule optimization using genetic algorithms*. In L. Davis, editor, Handbook of Genetic Algorithms , chapter 21, pages 332-349. Van Nostrand Reinhold, New York, 1991.