PARAMETER CONTROL IN MULTIRECOMBINATED EVOLUTIONARY ALGORITHMS FOR THE FLOW SHOP SCHEDULING PROBLEM

Vilanova G. Villagra A., Pandolfi D., De San Pedro M,
Proyecto UNPA-29/B032[1]
División Tecnología
Unidad Académica Caleta Olivia
Universidad Nacional de La Patagonia Austral
(9011) Caleta Olivia – Santa Cruz - Argentina
e-mail: gvilanov@satlink.com, avillagra@uaco.unpa.edu.ar,
dpandolfi@uaco.unpa.edu.ar, edesanpedro@hotmail.com
Phone/Fax : +54 297 4854888

Gallard R.
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)[2]
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
(5700) - San Luis – Rep. Argentina
e-mail: rgallard@unsl.edu.ar
Phone: + 54 652 20823 Fax : +54 652 30224

ABSTRACT

Improvements in evolutionary algorithms (EAs) consider multirecombination, allowing multiple crossover operations on a pair of parents (MCPC, *multiple crossovers per couple*) or on a set of multiple parents (MCMP, *multiple crossovers on multiple parents*). Evolutionary algorithms have been successfully applied to solve scheduling problems. MCMP-STUD and MCMP-SRI are novel MCMP variants, which considers the inclusion of a *stud-breeding individual* in a pool of random immigrant parents In this paper the proposal is to generate the *stud-breeding individual* by means of a robust conventional heuristic, the CDS. In a multirecombined EA, setting of parameters n_1 (number of crossovers) and n_2 (number of parents) remained as an open question. In previous works; they were empirically determined, or a deterministic rule was applied. In this paper self adaptation of parameters n_1 and n_2 is implemented, the idea is to code the parameters within the chromosome and undergo genetic operations. Hence it is expected that better parameter values be more intensively propagated.
The present paper discusses different multi-recombined methods and contrasts their performance when different parameter control methods are applied, to find the minimum makespan for selected instances of the FSSP.

KEYWORDS: Evolutionary algorithms, Multiple Crossovers, Multiple Parents, Scheduling, Parameter Control.

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

1. INTRODUCTION.

The *flow-shop sequencing problem* is generally described as follows: There are *m* machines and *n* jobs. Each job consists of *m* operations and each operation requires a different machine, all *n* jobs have to be processed in the same sequence on *m* machines. The processing time of each job in each machine is given. Frequently, the main objective is to find the sequence of jobs minimizing the maximum flow time, which is called the *makespan* [15]. The flow-shop problem has been proved to be NP-complete. Hence conventional and evolutionary heuristics have been developed by many researchers to solve the FSSP. In the category of conventional heuristics we can mention GUPTA, PALMER, NEH, and CDS. [10, 12, 13, 14, 15].

A new feature known as multirecombination applies several crossover operations on the set of (2 or more) parents. By means of multirecombination (MCPC and MCMP) better results were achieved. This implies higher quality of the best solution found throughout the evolutionary process, as well as an improved final population surrounding near optimal solutions. This later property also provides a sort of fault tolerance, because if eventually the dynamics of the system impedes using the best solution found then a better set of alternative solutions are available. The multirecombined methods were applied to FSSP [1] and contrasted on a series of suitable experiments against previous successful approaches of Tsujimura and Reeves. [18] Two variant MCMP-STUD [19] and MCMP-SRI [21] were recently proposed. Here, a stud (breeding individual) is selected for recombining with a subset of parents from the old population. The members of this mating pool subsequently undergo multiple crossover operations. In the case of MCMP-SRI, the stud (breeding individual) is generated by the CDS heuristics and the rest of the members of the parents pool are random immigrant.

Setting of parameters n_1 (number of crossovers) and n_2 (number of parents) in a multirecombined EA, remained as an open question in previous works; they were empirically determined or a deterministic rule was applied. Self adaptation is a new field in evolutionary computation which advises to dynamically update parameters of the algorithm by evolving them as part of the chromosome structure. Previous work of Spears [16] suggested adaptive approaches to select the type of crossover operator to be applied to each couple during an evolutionary algorithm execution.
In this paper self adaptation of parameters n_1 and n_2 is implemented, the idea is to codify the parameters within the chromosome and undergo genetic operations. Hence it is expected that better parameter values be more intensively propagated.

The following sections discuss these new multi-recombined methods, describe parameter control methods and contrast their performance when they are applied to find the minimum makespan for selected instances of the FSSP.

2. IMPROVED EVOLUTIONARY COMPUTATION APPROACHES

Multiple crossover per couple (MCPC) [6,7] is a novel crossover method. It was applied to optimize classic testing functions and some harder (non-linear, non-separable) functions. For each mating pair MCPC allows a variable number of children. It is possible to choose for insertion in the next generation the best, a randomly selected or all of the generated offspring. In those earlier works it was noticed that in some cases MCPC found better results than those provided by SCPC. Also a reduced running time resulted when the number of crossovers per couple increased, and best quality results were obtained allowing between 2 and 4 crossover per couple. However in some cases the method increased the risk of premature convergence due to a loss of genetic diversity. To overcome this problem further successful approaches were undertaken [8]. Moreover, seeking for exploitation of a greater sample from the problem space, the multi-recombination was extended and applied to a set of more than two parents.

In MCMP-STUD[19], a mating pool is created by selection of n_2 individuals from the old population. Then the parent with minimum makespan (stud) mates every other parent in the pool. At that time *partially mapped crossover* (PMX) is applied to each couple and from the new offspring, after eventual mutation, the best one is selected for insertion in the next generation. The members of this mating pool subsequently undergo multiple crossover operations.

In MCMP-SRI [21], the process for creating offspring is performed as follows (see figure 1). From the old population an individual, assumed as the stud, is selected by means of proportional selection. The number of n_2 parents in the mating pool is completed with randomly created individuals (random immigrants). The stud mates every other parent, the couples undergo crossover and $2*n_2$ offspring are created. The best of these $2*n_2$ offspring is stored in a temporary children pool. The crossover operation is repeated n_1 times, for different cut points each time, until the children pool is completed. Finally, the best offspring created from n_2 parents and n_1 crossover is inserted in the new population.

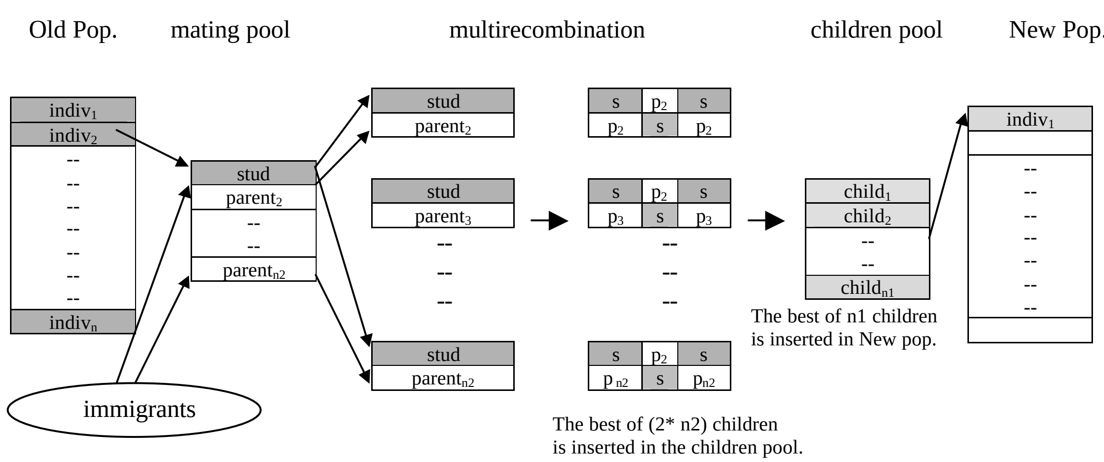

Fig. 1. The stud and random immigrants multirecombination process.

3. PARAMETER CONTROL

Today a great interest exists in methods including mechanisms to control parameters used by evolutionary algorithms during execution. Eiben, Hinterding and Michalewicz [7,9,11,12] gave the following main categories of parameter control:

- *Deterministic Parameter Control*: This is the case in which the parameter value is modified according with a deterministic rule, without any feedback of the searching process performed by the strategy.
- *Adaptive Parameter Control*: In this case some feedback information of the searching process is used to determine the direction and magnitude of the change in the parameters.
- *Self-adaptive Parameter Control*: Here the parameters to be adapted are coded within the chromosomes and undergo genetic operations. The best individuals of the population have better chances of survival and reproduction. Hence it is expected that better parameter values be more intensively propagated.

As n_1 (number of crossovers) and n_2 (number of parents) to be applied to a couple, are parameters of the algorithm which are included as part of an individual, our present approach appertains to the last above mentioned category. In that way we have three searching spaces: one corresponding to the objective function and the others associated to n_1 (number of crossovers) and n_2 (number of parents) respectively.

In *Deterministic Parameter Control (DPC)* [20], the parameter value is modified according to a deterministic rule based only on the current progress of the process indicated by the generation number. In the initial stages of the evolutionary process exploration is necessary while in the final stages exploitation of the relevant search space areas is advisable. Consequently, in our experiments, n_1 starts with a low value and then increases while n_2 starts with a high value and then decreases during the evolutionary process. The deterministic rule is a lineal function of the current generation number.

$$n_1 = integer\ (n_{1\ max} * (currentgeneration\ /maxgen)) + 1$$
$$n_2 = n_{1\ max} - n_1 + 1$$

In this way $n_{1\ max}$ determines dynamically both n_1 and n_2.

In the case of *Self Adaptation Parameter Control (SPC)* similar to previous implementations [8], n_1 and n_2 were encoded in two extra genes at the chromosome and these last two genes undergo *one-cut point crossover operation.*

4. EXPERIMENTS AND RESULTS

We tested four different approaches contrasting multirecombined methods when different parameter control methods were applied: MCPC-PMX, MCMP-CUSX, MCMP-STUD and MCMP-SRI with fixed n_1 and n_2 parameter setting (FPS). Then the same four methods were run with *Deterministic Parameter Control* (DPC) and *Self Adaptation Parameter Control (SPC)*.

It worth noting that CUSX (*controlled uniform scanning crossover*) is a variant of USX (*uniform scanning crossover)* which manage permutations ensuring the birth of feasible offspring.

Approaches with Fixed Parameter Setting (FPS)

1. MCPC-PMX, multiple crossovers per couple using PMX. $n_1 = 3$.
2. MCMP-(CUSX), multiple crossovers on multiple parents using USX [1] ($n_1=2$, $n_2=6$).
3. MCMP-STUD, multiple crossovers on multiple parents, the best designated as the stud, with PMX, ($n_1= 6$, $n_2=8$).
4. MCMP-SRI, multiple crossovers on multiple parents, the best designated as the stud, with PMX, ($n_1= 6$, $n_2=8$).

The fixed values n_1 and n_2 were selected after a large number of trials.

Approaches with Deterministic Parameter Control (DPC)

5. MCPC-PMX.
6. MCMP-CUSX.
7. MCMP-STUD.
8. MCMP-SRI

In all cases $n_{1\ max}$ was set to 6.

Approaches with Self Adaptation Parameter Control (SPC), 9,10,11,12 the same as in DPC but the first individual in the initial population in (MCPC-PMX, MCMP-CUSX), and the *stud-breeding individual* (in MCMP-STUD and MCMP-SRI) was generated by CDS heuristic.

Furthermore, every EA ran with a population of 100 individuals, elitism, maximum number of generations fixed at 100, and probabilities of crossover and mutation set to 0.65 and 0.3, respectively.

All approaches were tested for six Taillard´s benchmarks [17] for FSSP. We selected four instances for each of the following problem sizes: 20x5, 20x10, 20x20, 50x5, 50x10, and three instances for the 50x20 problem size. For each instance a series of ten runs was performed. As an indication of the performance of the algorithms the following variables were chosen:

Ebest: (Abs(*opt_val* – best value)/opt_val)*100. It is the percentile error of the best found individual in one run when compared with the known (or assumed) optimum value *opt_val*. It gives us a measure of how far the best individual is from that opt_val.

AvEbest: It is the average value of the error, over the total number of runs in one instance.

Mean AvEbest: It is the mean average value of the error, over the total number of runs and instances.

Next tables and figures show results obtained for all problem sizes under each approach. Here, the values for the *AvEbest* and *Mean AvEbest* from the corresponding selected instances and experiments are indicated.

Fixed Parameter Setting (FPS)

Instance	1. MCPC PMX	2. MCMP CUSX	3. MCMP Stud	4. MCMP SRI
20X5-1	1.667	4.593	1.651	2.293
20X5-2	0.935	2.664	0.809	1.001
20X5-3	3.043	11.175	7.095	6.883
20X5-4	2.196	9.536	5.553	6.721
Mean	**1.960**	**6.992**	**3.777**	**4.225**

Deterministic Parameter Control (DPC)

Instance	5. MCPC PMX	6. MCMP CUSX	7. MCMP Stud	8. MCMP SRI
20X5-1	3.693	1.956	1.549	3.036
20X5-2	1.744	1.376	1.163	1.354
20X5-3	6.568	4.274	5.680	8.326
20X5-4	6.102	4.022	5.553	8.213
Mean	**4.527**	**2.907**	**3.486**	**5.232**

Table 1. *AvEbest* values for 20 x 5 instances

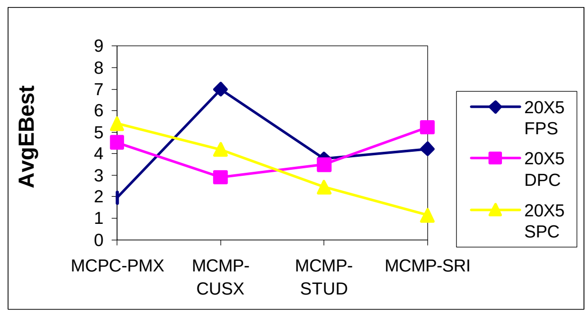

Fig. 1: Mean *AvEbest* for 20 x 5 instances

Self Adaptation Parameter Control (SPC)

Instance	9. MCPC PMX	10. MCMP CUSX	11. MCMP Stud	12. MCMP SRI
20X5-1	3.592	3.177	2.551	1.275
20X5-2	2.031	1.685	2.653	1.487
20X5-3	8.390	6.068	2.449	0.375
20X5-4	7.595	5.893	2.152	2.109
Mean	**5.402**	**4.206**	**2.451**	**<u>1.132</u>**

Table 1. (cont) *AvEbest* values for 20 x 5 instances

By observing table1 (cont.) and figure 1 we conclude that MCMP-SRI with SPC is the best performer, followed by MCMP-STUD with SPC. The worst performer was MCMP-CUSX with FPS showing that the method can be improved if parameter control is exerted. DPC and SPC degrades the performance of MCPC-PMX indicating that $n_{1\ max}$ is too high for this problem size. MCMP-STUD experiments a slight improvement in performance if DPC is applied compared with FPS, much better if SPC is applied.

Instance	1. MCPC PMX	2. MCMP CUSX	3. MCMP Stud	4. MCMP SRI
Fixed Parameter Setting (FPS)				
20X10-1	3.167	11.700	2.655	8.180
20X10-2	3.707	11.302	3.701	7.631
20X10-3	3.991	10.836	3.737	7.908
20X10-4	4.122	12.504	3.621	8.273
Mean	**3.747**	**11.585**	**3.429**	**7.998**
Deterministic Parameter Control (DPC)				
Instance	5. MCPC PMX	6. MCMP CUSX	7. MCMP Stud	8. MCMP SRI
20X10-1	6.941	4.893	4.949	8.723
20X10-2	5.980	5.986	5.750	7.860
20X10-3	6.604	5.568	5.321	9.559
20X10-4	7.380	5.493	5.283	10.239
Mean	**6.726**	**5.485**	**5.326**	**9.905**
Self Adaptation Parameter Control (SPC)				
Instance	9 MCPC PMX	10 MCMP CUSX	11 MCMP Stud	12. MCMP SRI
20X10-1	7.212	8.034	7.870	2.592
20X10-2	6.281	6.227	7.715	2.257
20X10-3	8.576	7.246	7.574	2.375
20X10-4	8.440	8.077	8.585	3.015
Mean	**7.627**	**7.396**	**7.936**	**2.559**

Table 2. *AvEbest* values for 20 x 10 instances

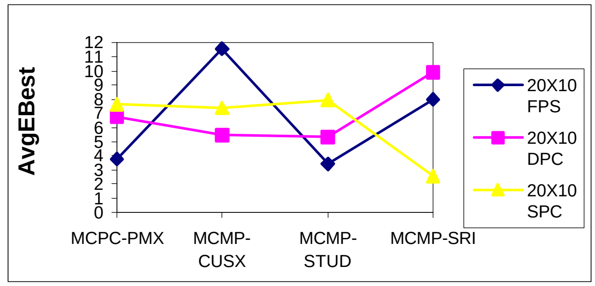

Fig. 2: Mean *AvEbest* for 20 x 10 instances

In table 2 and figure 2 we can see that MCMP-SRI with SPC is the best performer, followed by MCMP-STUD with FPS. The worst performer was again MCMP-CUSX with FPS showing that the method can be improved when parameter control is applied. DPC degrades the performance of MCPC-PMX, MCMP-STUD and MCMP-SRI indicating that $n_{1\ max}$ is too high for this problem size in the first case and that a loss of the inherent balance between exploration and exploitation can occur in the second case.

From table 3 and figure 3 we can notice that MCMP-STUD with either SPS or DPC are the best performers. As before, the worst performer is MCMP-CUSX with FPS, and the method improves when parameter control is applied. Also MCPC-PMX improves under DPC in this problem size.

Fixed Parameter Setting (FPS)				
Instance	1. MCPC PMX	2. MCMP CUSX	3. MCMP Stud	4. MCMP SRI
20X20-1	6.099	6.508	1.881	6.021
20X20-2	5.971	7.695	2.495	6.914
20X20-3	5.198	6.943	2.266	5.292
20X20-4	6.001	7.382	2.564	6.932
Mean	**5.817**	**7.132**	**2.301**	**6.290**
Deterministic Parameter Control (DPC)				
Instance	5. MCPC PMX	6. MCMP CUSX	7. MCMP Stud	8. MCMP SRI
20X20-1	4.619	3.805	4.362	7.314
20X20-2	4.552	4.995	4.686	8.052
20X20-3	4.140	4.076	4.037	6.758
20X20-4	4.669	5.083	4.148	7.526
Mean	**4.495**	**4.490**	**4.308**	**7.413**
Self Adaptation Parameter Control (SPC)				
Instance	9 MCPC PMX	10 MCMP CUSX	11 MCMP Stud	12. MCMP SRI
20X20-1	5.211	6.143	2.630	6.670
20X20-2	5.929	5.914	2.329	7.090
20X20-3	4.484	5.997	2.190	5.924
20X20-4	5.016	6.253	2.773	6.217
Mean	**5.160**	**6.077**	**2.481**	**6.475**

Table 3. *AvEbest* values for 20 x 20 instances

Fixed Parameter Setting (FPS)				
Instance	1. MCPC PMX	2. MCMP CUSX	3. MCMP Stud	4. MCMP SRI
50X5-1	0.914	2.533	0.400	1.836
50X5-2	2.350	4.718	1.334	4.478
50X5-3	1.709	4.243	1.328	3.636
50X5-4	2.708	4.878	1.109	4.507
Mean	**1.920**	**4.093**	**1.043**	**3.614**
Deterministic Parameter Control (DPC)				
Instance	5. MCPC PMX	6. MCMP CUSX	7. MCMP Stud	8. MCMP SRI
50X5-1	1.252	1.072	0.844	2.173
50X5-2	1.941	2.297	1.595	4.682
50X5-3	2.415	1.740	1.381	3.705
50X5-4	3.144	1.854	1.516	4.547
Mean	**2.188**	**1.741**	**1.334**	**3.777**
Self Adaptation Parameter Control (SPC)				
Instance	9 MCPC PMX	10 MCMP CUSX	11 MCMP Stud	12. MCMP SRI
50X5-1	1.098	1.292	0.224	1.630
50X5-2	1.757	1.292	0.209	3.592
50X5-3	2.304	1.222	0.282	3.167
50X5-4	3.297	1.259	0.851	4.039
Mean	**2.114**	**1.468**	**<u>0.392</u>**	**3.107**

Table 4. *AvEbest* values for 50 x 5 instances

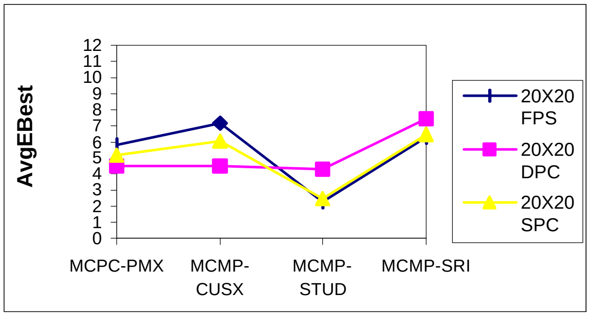

Fig. 3: Mean *AvEbest* for 20 x 20 instances

Fixed Parameter Setting (FPS)				
Instance	1. MCPC PMX	2. MCMP CUSX	3. MCMP Stud	4. MCMP SRI
50X10-1	9.150	13.478	3.950	11.478
50X10-2	9.419	12.766	4.464	12.521
50X10-3	11.512	15.119	4.892	14.085
50X10-4	8.267	12.001	3.642	10.826
Mean	**9.587**	**13.341**	**4.237**	**12.227**
Deterministic Parameter Control (DPC)				
Instance	5. MCPC PMX	6. MCMP CUSX	7. MCMP Stud	8. MCMP SRI
50X10-1	5.977	6.003	5.137	11.600
50X10-2	7.272	7.503	5.439	12.656
50X10-3	7.898	8.666	5.705	14.022
50X10-4	6.191	6.090	4.184	11.622
Mean	**6.834**	**7.066**	**5.116**	**12.475**
Self Adaptation Parameter Control (SPC)				
Instance	9. MCPC PMX	10. MCMP CUSX	11. MCMP Stud	12. MCMP SRI
50X10-1	7.160	10.446	3.395	11.392
50X10-2	8.316	10.650	4.060	12.172
50X10-3	8.530	11.198	3.970	12.933
50X10-4	6.687	9.191	4.885	10.082
Mean	**7.673**	**10.371**	**<u>4.077</u>**	**11.644**

Table 5. *AvEbest* values for 50 x 10 instances

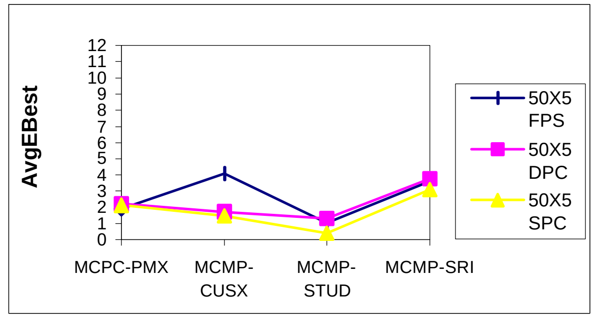

Fig. 4: Mean *AvEbest* for 50 x 5 instances

In table 4 and figure 4 we can observe that MCMP-STUD with FPS, DPC or SPC are the best performers with similar average error values. The worst performer was again MCMP-CUSX with FPS, and the method improves when parameter control is applied. DPC slightly degrades the performance of both MCPC-PMX and MCMP-STUD.

Fixed Parameter Setting (FPS)				
Instance	1. MCPC PMX	2. MCMP CUSX	3. MCMP Stud	4. MCMP SRI
50X20-1	9.257	14.379	3.853	12.895
50X20-2	10.250	14.748	5.467	13.529
50X20-3	10.701	15.379	5.782	14.119
Mean	**10.069**	**14.835**	**5.034**	**13.514**
Deterministic Parameter Control (DPC)				
Instance	5. MCPC PMX	6. MCMP CUSX	7. MCMP Stud	8. MCMP SRI
50X20-1	7.855	7.425	5.471	13.283
50X20-2	8.861	9.276	6.923	13.812
50X20-3	8.307	8.664	6.069	14.016
Mean	**8.341**	**8.455**	**6.154**	**13.703**
Self Adaptation Parameter Control (SPC)				
Instance	9. MCPC PMX	10. MCMP CUSX	11. MCMP Stud	12. MCMP SRI
50X20-1	6.689	10.945	4.021	12.862
50X20-2	8.248	11.456	3.786	13.241
50X20-3	8.037	10.990	5.879	13.604
Mean	**7.658**	**11.130**	**4.562**	**13.236**

Table 6. *AvEbest* values for 50 x 20 instances

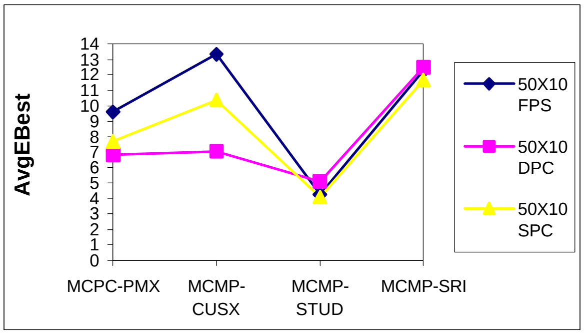

Fig. 5: Mean *AvEbest* for 50 x 10 instances

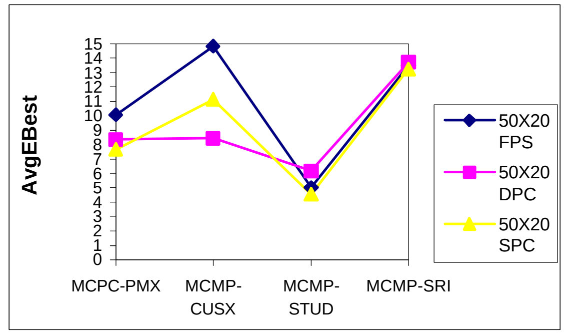

Fig. 6: Mean *AvEbest* for 50 x 20 instances

From table 5 and figure 5 we can see that again MCMP-STUD with FPS, DPC or FPC are the best performers. The worst performer was again MCMP-CUSX with FPS, and the method improves when parameter control is applied. Both MCPC-PMX and MCMP-CUSX significantly improve under DPC and SPC in this problem size.

A similar situation to the one in the last case, can be observed in table 6 and figure 6 for this problem size. Again MCMP-STUD with FPS, DPC and SPC are the best performers. MCMP-CUSX with FPS remains as the worst performer. Both MCPC-PMX and MCMP-CUSX improve under DPC and SPC in this problem size, and more significantly for the latter.

5. CONCLUSIONS

This paper introduced a parameter control method, the *Self Adaptation Parameter Control*, in multi-recombined evolutionary algorithms for the Flow Shop Scheduling Problem. In *Self Adaptation Parameter Control* the idea is to code the parameters n_1 (number of crossovers) and n_2 (number of parents) within the chromosome and undergo genetic operations. Hence it is expected that better parameter values be more intensively propagated.Results achieved are contrasted against results obtained in previous work under other parameter control method, the *Deterministic Parameter Control*.

By analyzing results we can remark:

For any of the considered performance variables MCMP-CUSX with deterministic parameter control (DPC) is the most sensitive method to DPC for any problem size. It is followed by MCPC-PMX, which shows improvements in most cases, except in the smaller problem sizes. In all cases MCMP-STUD and MCMP-SRI improve their performance, and much better in smaller problem size. These different behaviours can be explained as follows. Results for FPS were obtained after a number of trials to find better parameter setting while for these preliminary tests with DPS all the algorithms ran with the same $n_{1\ max}$ value. When SPC was applied a robust conventional heuristic, the CDS to generate one individual in the initial population for MCPC-PMX and MCMP-CUSX and as a stud-breeding-individual for MCMP-STUD and MCMP-SRI. Consequently:

- MCPC-PMX was affected in the smaller problem sizes for excessive exploitation due to a high maximum number of crossover leading to premature convergence. As a consequence in some smaller problem sizes no improvements or even degradation can be expected.

- MCMP-CUSX is intrinsically affected for a crossover method which preventing unfeasible offspring reduces the searching abilities of the algorithm. Consequently, improvements are to be expected in many cases.
- MCMP-STUD and MCMP-SRI are methods, which inherently balance exploration and exploitation in the searching space when parameters are adequately selected so as a consequence of applied self adaptation better results were obtained.

6. ACKNOWLEDGEMENTS

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

7. REFERENCES

1. Bain M., Ferreyra J., Pires A., Varas V., Vilanova G., Gallard R.H, ¨Enhanced Evolutionary Approaches To The Flow Shop Scheduling Problem¨, Second Interna-tional ICSC Symposium of Intelligent Systems EIS 2000, June 27-30 , 2000, University of Paisley, Scotland U.K.

2. Davis, L. Applying adaptive algorithms to domains. In proceeding of the International Joint Conference on Artificial Intelligence. pp. 162-164. 1985.

3. 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

4. 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.

5. Eiben A. E., Hinterding R.,Michalewicz Z. – *Parameter Control in Evolutionary Algorithms*, IEEE Transactions on Evolutionary Computation, Vol. 3 No.2, 1.999.

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), pp 103-106, ISBN 0-7803-3949-5, Indianapolis, USA, April 1997.
7. 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.

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), pp 419-425. Universidad Nacional del Comahue, Argentina, Octubre 1998.

9. Grefenstette J.J., *Optimization of control parameters for genetic Algorithms*, IEEE Transactions on Systems, Many and Cybernetics, 16 (1):122-128 1988.

10. Gupta J., *A functional heuristic algorithm for the flowshop scheduling problem*, Operational Research Quarterly 22, 1971,39-48.

11. Hinterding R., Michalewicz Z., Eiben A. E. – *Adaptation in Evolutionary Computation – A survey,* Proceedings of the 4th IEEE International Conference on Evolutionary Computation, Indianapolis, April 13-16, 1.997, pp. 65-69.

12. Michalewicz, M., *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, third revised edition, 1996.

13. Morton T., Pentico D., *Heuristic scheduling systems*, Wiley series in Engineering and technology management. John Wiley and Sons, INC, 1993.

14. Oliver I. M., Smith D. J., Holland J. R. C. – *A Study of Permutation Crossover operators on the Travelling Salesman Problem* - Proceedings of the 2nd Int. Conf. on Genetic Algorithms, pp 224-230. Lawrence Erlbaum Assoc., NJ, 1987.

15. Pinedo Michael (1995) – *Scheduling- Theory , Algorithms, and Systems*. Prentice Hall International in Industrial and System Engineering.

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

17. Taillard, E. *Benchmarks for basic scheduling problems*, European Journal of Operational Research, vol.64,pp.278-285,1993. http://wwww.idsia.ch/~eric/problems.dir/ordonnancement.dir/ordonnancement.html).

18. Tsujimura Y., Gen M., Kubota E., *Flow shop scheduling with fuzzy processing time using genetic algorithms.* The 11th Fuzzy Systems Symposium pp 248-252. Okinawa. 1995.

19. Vilanova G., Pandolfi D., De San Pedro M., Villagra A., Gallard R. Multiple recombination and breeding in evolutionary algorithm for flow shop scheduling problems. Proceedings of the International Conference in Soft Computing. University of Paisley, Scotland, U.K., June2001.

20. Vilanova G., Pandolfi D., De San Pedro M., Villagra A., Bain M, Gallard R. DERMINISTIC PARAMETER CONTROL IN MULTIRECOMBINATED EVOLUTIONARY ALGORITHMS FOR THE FLOW SHOP SCHEDULING PROBLEM . Proceedings of the SCI 2001 Orlando. USA, June 2001.

21. Vilanova G., Pandolfi D., De San Pedro M., Villagra A., Gallard R.*Solving the single-machine common due date problem via studs and inmigrants in evolutionary algorithms*. Proceedings of the SCI 2001 Orlando. USA, June 2001.