INCEST PREVENTION AND MULTIRECOMBINATION IN EVOLUTIONARY ALGORITHMS TO DEAL WITH THE FLOW SHOP SCHEDULING PROBLEM

BAIN M., VILANOVA G.
Proyecto UNPA-29/B017[1]
División Tecnología
Unidad Académica Caleta Olivia
Universidad Nacional de La Patagonia Austral
Ruta 3 Acceso Norte s/n
(9011) Caleta Olivia – Santa Cruz - Argentina
e-mail: { mebain@infovia.com.ar,gvilanov@satlink.com }
Phone: +54 297 4550654 / 854888
Fax : +54 297 854888 int 105

GALLARD R.
Proyecto UNSL-338403[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

Scheduling concerns the allocation of limited resources for tasks over time. It is a process of making decisions that has, as a goal, the optimization of one or more objectives. Frequently, the main objective to be minimized is the completion time of the last job to abandon the system, which is called *makespan*.

In many production systems a number of operations must be done on every job and often these operations have to be done in the same order on all jobs. This scheduling approach is known as the Flow Shop Scheduling Problem (FSSP). The present paper discusses the new multi-recombinative method and shows the performance of enhanced evolutionary approaches under permutation representation combined with a successfull previous approach proposed by another researchers, the extended incest prevention (EIP), consist of maintaining information about ancestors within the chromosome and modifying the selection for reproduction in order to impede mating of individuals belonging to the same “family”, for a predefined number of generations.

Results of the methods proposed here are contrasted with those obtained under previous evolutionary approaches to the FSSP.

KEYWORDS: Evolutionary algorithms, genetic diversity, premature convergence, incest prevention.

---

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

1. INTRODUCTION.

Scheduling is not only present in most manufacturing and production systems but in most information processing environments as well. It also exists in transportation and distribution setting and in other types of service industries. In the scheduling problem resources and tasks may take many forms. Resources might be machines in a workshop, runways at an airport, processing units in a computing environment, and so on. Tasks might be operations in a production process, take-off and landings at an airport, stages in a construction project, execution of computer programs and so on.

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, so n jobs have to be processed in the same sequence on m machines. The processing time of each job in each machine is known. The objective is to find the sequence of jobs minimizing the maximum flow time which is called *makespan* [17]. The flow-shop problem has been proved to be NP-complete. Hence conventional and evolutionary heuristics have been developed by many researchers [2, 11, 13, 14, 16, 18, 20] to provide a good and fast solution. This problem has been proved as NP-hard for even a very small number of resources. Conventional heuristics and evolutionary algorithms (EAs) have been developed by many researchers [12,13,18,20] to solve flow-shop problems.

Evolutionary Computation [12], as an emergent research field, which provides new heuristics to problem optimization where traditional approaches make the problem computationally intractable, it is continuously showing its own evolution and enhanced approaches include latest multi-recombinative method involving *multiple crossovers per couple* (MCPC). Important research has been made in evolutionary computation, to maintain a good balance between exploration and exploitation of solutions while searching in a problem space. Extreme exploitation can lead to premature convergence (loss of population diversity before optimal or at least satisfactory values have been found) and intense exploration can make the search ineffective. [12]. Eshelman and Shaffer [5] attempted to maintain population diversity by using diverse strategies focusing on mating, recombination and replacement. One of their approaches, called incest prevention, avoided mating of pairs showing similarities based on the parent's hamming distance.

In this work deepening research reported in [3], is implemented with the extended approach of incest prevention proposed in [1].

2. EVOLUTIONARY ALGORITHMS IN SCHEDULING PROBLEMS.

Evolutionary algorithms (EAs) have been successfully applied to solve flow-shop problems. Tsujimura et al [20] provided evidence of the performance of genetic algorithms (GAs) contrasted with conventional approaches using well known crossover operators such as *partially-mapped crossover* (PMX) [10], *order crossover* (OX) [4] and *cycle crossover* (CX), [15]. Because of the flow-shop problem is essentially a permutation schedule problem, a permutation can be used as the representation scheme of chromosomes, which is the natural one for a sequencing problem. For example, let the k[th] chromosome be: $v_k = [10\ 12\ 9\ 11]$. This means that the job sequence is $j_{10}, j_{12}, j_9, j_{11}$. The permutation representation, also called *order representation*, may lead to illegal offspring if the traditional one-point crossover operator is used. Consequently, during the past decades, several crossover operators have been proposed for permutation representation, such as the above mentioned PMX, OX and CX.

Reeves [18] proposed a hybrid approach, which inserts a chromosome as a seed in the initial population generated by the NEH heuristic algorithm [14]. He suggested genetic operators in his implementation what he called *one-cut-point crossover* (OCPX). It consist of choosing one-cut-point randomly, and then taking the pre-cut section of the first parent and filling up the offspring by taking in the order they appear legitimate genes from the second parent. He tried two types of mutation, one of them, an *exchange mutation,* which was a simple exchange of two genes of the chromosome, chosen at random. The other, a *shift mutation* was a shifting of one gene (chosen randomly) to the right or left a random number of places. After a few experiments, Reeves observed that shift mutation seemed to be better than exchange. Reeves tested his GA on Taillard's benchmarks [19] and concluded that simulated annealing algorithms and GAs produce comparable results for the *flow-shop sequencing* problem for most sizes and types of problems, but GAs perform relatively better for large problems and reach a near-optimal solution more quickly.

3. A MULTI-RECOMBINATIVE APPROACH

As we said extreme exploitation can lead to premature convergence and intense exploration can make the search ineffective. [12] The intuition behind the applicability of the crossover operator is information exchange between different potential solutions. In EAs the common approach is to operate once on each mating pair after selection. Such procedure is known as the SCPC (Single Crossover Per Couple) approach. 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 to consider is; how would the performance of an evolutionary program be affected by the use of a multiple crossovers per couple operation?. Performance improvement was observed when MCPC was implemented in [3]. Next subsections explain this multi-recombinative approach in more detail.

3.1 THE MULTIPLE CROSSOVER PER COUPLE APPROACH (MCPC)

Multiple crossover per couple (MCPC) [6] is a newly introduced 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 created 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 by using self–adaptation of MCPC parameters, [9] and by combining MCPC with an alternative selection method; *fitness proportional couple selection* (FPCS) [7], which first creates an intermediate population of couples where both individuals were chosen by proportional selection. Then a criterion is applied to establish the fitness of a couple and subsequently, couples are selected for crossing-over based on couple fitness.

4. I NCEST PREVENTION

In the case of multimodal functions the problem space, also called the *fitness landscape*, provide multiple suboptimal points. Depending on the type of operators used and their frequency of application, the convergence to these suboptimal points can arise. This effect, known as premature convergence, is mainly derived from a loss of population diversity before optimal, or at least satisfactory values, have been found.

A possible strategy to maintain population (genetic) diversity, attempting to avoid premature convergence is a mating strategy known as *incest prevention*. This approach was first used by Eshelman and Schaffer [5] who avoided mating of those pairs showing similarities. As a bit string representation was used for their experiments similarities were determined on the parent's hamming distance. An extended, representation-independent-approach of incest prevention: *extended incest prevention* (EIP) was proposed in [1] by maintaining information about ancestors within the chromosome structure and modifying the selection for reproduction. In this way mating of individuals belonging to the same "family" is avoided for a predefined number of generations. As it is reported this novel approach was tested on a set of multimodal functions and showed evidence of better performance when contrasted with traditional GA approaches on optimization of multimodal functions of varied difficulty. The optimal value was reached in many runs of each series and the mean value for the best individual throughout the series was always found earlier and was better than with a simple genetic algorithm (SGA).

5. EXPERIMENTS AND RESULTS

According to Tsujimura's and Reeves's works we tested 8 different approaches contrasting the conventional (simple crossover per couple SCPC), multiple crossover per couple (MCPC) and variants of them including Incest prevention (I). They were:

- *SCPC-T*, Tsujimura's approach. Uses OX, CX or PMX.
- *SCPC-R*, Reeves's approach. Uses OCPX and insertion of seeds in the initial population.
- *SCPC-TI*, Tsujimura's approach with incest prevention.
- *SCPC-RI*, Reeves's approach with incest prevention.
- *MCPC-T*, Multiple crossovers per couple using OX, CX or PMX.
- *MCPC-R*, Multiple crossovers per couple using OCPX and insertion of seed in the initial population.
- *MCPC-TI*, Multiple crossovers per couple. Tsujimura's approach with incest prevention.
- *MCPC-RI*, Multiple crossovers per couple Reeves's approach with incest prevention.

All approaches were tested for five Taillard's benchmarks [19] for the flow shop problem. We selected the following problem sizes: 20x5, 20x10, 20x20, 50x5, 50x10. A total of 36 different experiments were designed. For each instance a series of five runs was performed. Experiments consisted of varying parameters such as the number of crossover n_1, the number of offspring to be inserted in the next generation n_3, the number of seeds used in the initial population n_4, and the type of crossover used. The number of parents n_2,was fixed at 2. Besides all the EAs used the following parameter settings:

Population size	100 /120 (Incest Prevention)
Crossover Probability	0.65
Mutation Probability	0.01
Maximum No. of Generations	100
Elitism	Yes

As an indication of the performance of the algorithm the following relevant variables were chosen:

Ebest : It is the percentile error of the best found individual when compared with the benchmark upper bound for the optimal makespan. It gives us a measure of how far we are from that upper bound.

Epop : It is the percentile error of the population mean fitness when compared with benchmark upper bound for the optimal makespan. It tells us how far the average individual is from that upper bound makespan benchmark.
Gbest : Identifies the generation where the best valued individual (retained by elitism) was found.

Tables 1, 2 and 3 show the results obtained for the considered performance variables in three distinct problem sizes under each approach. The values for parameters n_1 to n_4 shown here conform the best combination found during the trials. The optimal parameter configuration for the experiments was: number of crossover *n1*, between 3 and 5 (for the multirecombinative approach), the number of offspring to be inserted in the next generation *n3* was fixed at 1; the number of seeds used in the initial population (Reeves's approach) was 4.
In the next tables mean values for the performance variables from the corresponding selected instances and experiments are indicated. Boldfaced values are the best achieved.

Approach	Instances: 20 x 5							
	Parameters				xover method	Performance variables		
	n_1	n_2	n_3	n_4		Ebest	Epop	Gbest
SCPC-T	1	2	1	0	PMX	2.1	2.7	35
SCPC-R	1	2	1	1	OCPX	7.0	9.9	**11.2**
SCPC-TI	1	2	1	0	PMX	**1.5**	**1.6**	43.8
SCPC-RI	1	2	1	1	OCPX	3.5	4.6	24.8
MCPC-T	4	2	1	0	PMX	4.8	10.2	13.3
MCPC-R	3	2	1	4	OCPX	6.0	9.2	14.7
MCPC-TI	4	2	1	0	PMX	**1.6**	**1.6**	34.9
MCPC-RI	3	2	1	4	OCPX	4.8	6.0	24.4

Table 1. Mean values of performance variables from four 20 x 5 instances

Approach	Instances: 20 x 20							
	Parameters				Xover method	Performance variables		
	n_1	n_2	n_3	n_4		Ebest	Epop	Gbest
SCPC-T	1	2	1	0	PMX	3.4	3.6	75.8
SCPC-R	1	2	1	4	OCPX	9.5	9.8	33.2
SCPC-TI	1	2	1	0	PMX	3.5	4.0	72.8
SCPC-RI	1	2	1	4	OCPX	8.2	10.0	28.4
MCPC-T	5	2	1	0	PMX	**2.5**	**2.5**	52.8
MCPC-R	4	2	1	4	OCPX	7.2	7.4	**23.8**
MCPC-TI	5	2	1	0	PMX	**2.8**	**2.8**	41.9
MCPC-RI	4	2	1	4	OCPX	5.7	6.0	44.5

Table 2. Mean values of performance variables from four 20 x 20 instances

Approach	Instances: 50 x 5							
	Parameters				Xover method	Performance variables		
	n_1	n_2	n_3	n_4		Ebest	Epop	Gbest
SCPC-T	1	2	1	0	OX	2.5	12.2	76.6
SCPC-R	1	2	1	1	OCPX	2.4	4.1	26.0
SCPC-TI	1	2	1	0	OX	2.9	12.9	80.8
SCPC-RI	1	2	1	1	OCPX	2.0	4.5	**13.0**
MCPC-T	5	2	1	0	PMX	3.4	3.4	25.2
MCPC-R	4	2	1	4	OCPX	**1.4**	2.1	88.6
MCPC-TI	5	2	1	0	PMX	1.5	7.7	65.6
MCPC-RI	4	2	1	4	OCPX	**0.8**	0.9	29.2

Table 3. Mean values of performance variables from four 50 x 5 instances

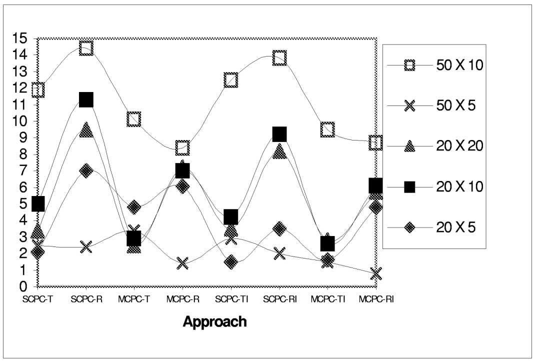

Fig 1. Mean Ebest values under each approach in different problem-size instances.

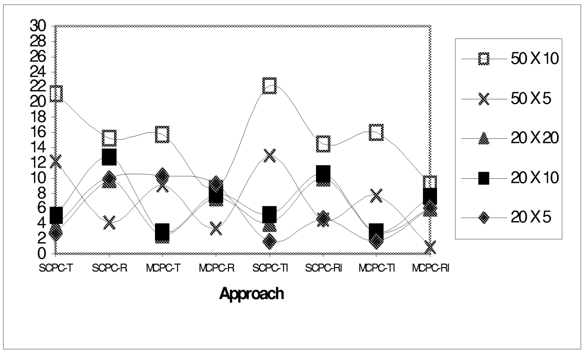

Fig 2. Mean Epop values under each approach in different problem-size instances.

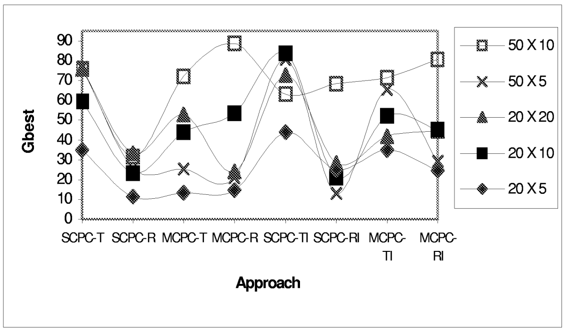

Fig 3. Mean Gbest values under each approach in different problem-size instances.

As additional analysis of results not shown in this presentation indicate that On all approaches using Tsujimura proposal, PMX outperforms CX and OX.

From table 1,2 and 3 we summarize:

	Problem size		
	20 X5	20X20	50x5
1	SCPC-TI	MCPC-T	MCPC-RI
2	MCPC-TI	MCPC-TI	MCPC-R
3	SCPC-T	SCPC-T	MCPC-TI
4	SCPC-RI	SCPC-TI	SCPC-RI
5	MCPC-T	MCPC-RI	SCPC-R
6	MCPC-RI	MCPC-R	SCPC-T
7	MCPC-R	SCPC-RI	SCPC-TI
8	SCPC-R	SCPC-R	MCPC-T

Table 4. Mean Ebest values in decreasing order

From table 4, we can conclude that for small problem sizes, (SCPC) with (EIP) was improved respect to (MCPC) for Tsujimura's approach, is not the same case for Reeves's approach but as long as the ratio n/m decreased, MCPC-I is better than MCPC in all cases.

Figures 1, 2 and 3, show the global results of this work. Here we can detect that:

- SCPC-R is the worst performer with mean *Ebest* values ranging from 2.4% to 14.4% but the methos is improved when incest prevention is applied under SCPC-RI ranging now from 2.0% to 13.8%.

- Best performers are MCPC-RI and MCPC-R with mean *Ebest* values ranging from 0.8% to 8.7% and 1.4% to 8.4%, respectively.

- Regarding *Epop*, also MCPC-RI and MCPC-R are the best performers with mean values ranging from 0.9% to 9.2% and 3.4% to 9.2%, respectively. This gives an indication of a final population more centred on the best found individual.

- Regarding *Gbest*, we conclude that as larger is the problem size, all methods require more generations to find the best individual. In the case of MCMP-R and MCMP-RI they require from 14 to 89 and 24 to 81 generations, respectively. Independently of the problem size, due to augmented genetic diversity, the multirecombinative approach needs a greater number of generations to find the best individual.

5. Conclusions.

This contribution introduces a multirecombinative approach, MCPC [6] combined with EIP (Extended incest prevention) [1], applied to the Flow Shop Scheduling Problem. These novel variant was contrasted on a series of suitable experiments against previous successful approaches of Tsujimura and Reeves. Better results are achieved, on the selected set of instances, by means of a multirecombinative approach and by both multirecombination (MCPC) jointly applied with incest prevention (EIP). This implies higher quality of solutions found throughout the evolutionary process, as well as an improved final population surrounding near optimal solutions. This later feature 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.

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 which we receive continuous support from.

7. References

[1] Alfonso H., Cesan P., Fernandez N., Minetti G., Salto C., Velazco L., *Improving Evolutionary Algorithms performance by Extending Incest Prevention.* Proceedings of the 4th Congreso Argentino de Ciencias de la Computación (CACiC'98). Universidad Nacional del Comahue, Argentina, Octubre 1998.

[2] Campbell H., Dudek R., Smith M., *A heuristic algorithm for the n job m machine sequencing problem*, Management Science 16, 1970, 630-637.

[3] Bain M., Ferreyra J.,Pires A., Varas V., Vilanova G., Gallard R., - Enhanced Evolutionary Approaches To the Flow Shop Scheduling Problem, Proceedings of the Second ICSC Symposium on Engineering of Intelligent Systems EIS2000, June 27-30, 2000, pp 439-444 University of Paisley, Scotland, U.K. ISBN 3-906454-23-1.

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

[5] Eshelman L. J, Schaffer J.D. Preventing Premature Convergence in Genetic Algorithms by Preventing Incest, Proceedings of the Fourth International Conference on Genetic Algorithms, pp 115-122, Morgan Kauffman, San Mateo California, USA, 1991.

[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 A., Gallard R.: *Couple Fitness Based Selection with Multiple Crossover per Couple in Genetic Algorithms.* Proceedings of the International Symposium on Engineering of Intelligent Sy

[8] stems (EIS´98), La Laguna, Tenerife, Spain Vol. 1, pp 235-241, ed. E.Alpaydin. Publishesd by ICSC Academic Press, Canada/Switzerland, February 1998.

[9] 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.

[10] 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.

[11] Goldberg D. E., Lingle R. – *Alleles, Loci and the TSP* – Proceedings of the First International Conference on Genetic Algorithms, Lawrence Erlbaum Assoc., NJ, 1985.

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

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

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

[15] Nawaz M., Enscore E., Ham I., *A heuristic algorithm for the m-machine n-job flow shop sequencing problem.* Omega vol II, pp 11-95, 1983.

[16] 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.

[17] Palmer D., *Sequencing jobs through a multistage process in the minimum total time*- A quick method of obtaining a near optimum, Operational Research Quarterly 16, 101-107, 1965.

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

[19] Reeves C., *A genetic algorithm for flow shop sequencing*, Computers and Operations Research, vol 22, pp5-13, 1995.

[20] 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).

[21] 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.