Evolutionary Optimization of Due Date Based Objectives in Unrestricted Identical Parallel Machine Scheduling Problems

Ferretti E., Esquivel S., Gallard R.

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)[1]
Facultad de Ciencias Físico-Matemáticas y Naturales
Universidad Nacional de San Luis
Ejército de los Andes 950 – Local 106
5700 – San Luis – Argentina
{ferretti, esquivel}@unsl.edu.ar

Abstract

Parallel machine scheduling, involves the allocation of jobs to the system resources (a bank of machines in parallel). A basic model consisting of *m* machines and *n* jobs is the foundation of more complex models. Here, jobs are allocated according to resource availability following some allocation rule. In the specialised literature, minimisation of the *makespan* has been extensively approached and benchmarks can be easily found. This is not the case for other important objectives such as the *maximum tardiness* and the *number of tardy jobs*. These problems are NP-hard for $2 \leq m \leq n$, and conventional heuristics and evolutionary algorithms (EAs) have been developed to provide acceptable schedules as solutions. To solve the unrestricted identical parallel machine scheduling problems, this paper proposes MCMP-SRI and MCMP-SRSI, which are two multirecombination schemes that combine studs, random and seed immigrants. Evidence of the improved behaviour of the EAs when inserting problem-specific knowledge is provided. Experiments and results are discussed.

Keywords: Parallel machine scheduling, evolutionary algorithms, multirecombination, maximum tardiness, number of tardy jobs.

Workshop: Agentes y Sistemas Inteligentes

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

1. Introduction

Unrestricted identical parallel machine scheduling problems are frequent in production systems. The completion time of the last job to leave the system, known as makespan (C_{max}), is one of the most common objective functions to be minimised. In a production system it is also usual to stress minimisation of others due date based objectives such as maximum tardiness (T_{max}) and number of tardy jobs (N_t).

These problems types have received considerable attention by different researchers. For many years their computational complexity remained as an open research topic until established as NP-Hard [16].

To provide reasonably good solutions in very short time the scheduling literature offers a set of dispatching rules and heuristics. Depending on the particular instance of the problem we are facing, some heuristics behave better than others. Among other heuristics [12], evolutionary algorithms (EAs) have been successfully applied to solve scheduling problems [9-11]. Current trends in evolutionary algorithms make use of multiparent [3-5] and multirecombined approaches [6-8]. This latter approach is known as *multiple crossovers on multiple parents* (MCMP). Instead of applying crossover once on a pair of parents, this scheme applies n_1 crossover operations on a set of n_2 parents. In order to improve the trade-off between exploration and exploitation in the search process a variant called MCMP-SRI [13,14] recombines a breeding individual (stud) by repeatedly mating individuals that randomly immigrate to a mating pool. Under this approach the random immigrants and the multi-mating operation with the stud incorporate exploration and exploitation, respectively in the search process.

If we are trying to incorporate knowledge to the blind evolutionary search process, the main issue here is how to introduce problem-specific knowledge? If optimality conditions for the solutions are known in advance we can restrict the search operating only on solutions which hold these conditions. When optimality conditions are unknown, which is the case here, one of the options is to import this knowledge from solutions that come out of heuristics specifically designed for the problem under consideration. These types of knowledge-based intermediate solutions contain some of the features included in the best (optimal or quasi-optimal) solution at the end of the evolutionary process.

Consequently MCMP-SRSI, a latest variant of MCMP-SRI, considers the inclusion of a stud-breeding individual in a pool of random and seed-immigrant parents. Here the seeds generated by conventional heuristics introduce the problem-specific knowledge. The following sections describe the above mentioned scheduling problems, ways of inserting *problem-specific knowledge* and provide a discussion of the results obtained.

2. Scheduling problems

The problems we are facing [16] can be stated as follows: n jobs are processed without interruption on some of the m equal machines belonging to the system; each machine can handle no more than one job at a time. Job j (j=1,...,n) becomes available for processing at time zero, requires an uninterrupted positive processing time p_j on a machine, and has a due date d_j by which it should ideally be finished. For a given processing order of the jobs, the earliest completion time C_j and the tardiness $T_j = \max\{C_j - d_j, 0\}$ of job j can readily be computed. The problem is to find a processing order of the jobs with minimum objective values. The objectives to be minimized are:

Maximum Tardiness : $T_{\max} = \max_j(T_j)$

Number of Tardy Jobs: $N_t = \sum_{j=1}^{n} \delta(T_j)$, where $\delta(T_j) = 1$ if $T_j > 0$

$\delta(T_j) = 0$ otherwise

3. Conventional approaches to scheduling problems

Dispatching heuristics assign a priority index to every job in a waiting queue. The one with the highest priority is selected to be processed next. There are different heuristics [12] for the above mentioned problems whose principal property is not only the quality of the results but also to give a schedule of the jobs close to the optimal sequence. The following dispatching rules and heuristics were selected to determine priorities, build schedules and contrast their outcomes with those obtained by the evolutionary algorithms proposed.

SPT (Shortest Processing Time first): The job with the shortest processing time is selected first. The final scheduled jobs are ordered satisfying: $p_1 \leq p_2 \leq \ ... \leq p_n$.

WSPT (Weighted Shortest Processing Time first): The job with the weighted shortest processing time is selected first. The final scheduled jobs are ordered satisfying:
$(w_1 / p_1) \geq (w_2 / p_2) \geq \ ... \geq (w_n / p_n)$.

EDD (Earliest Due Date first): The job with earliest due date is selected first. The final scheduled jobs are ordered satisfying: $d_1 \leq d_2 \leq \ \ldots \leq d_n$.

SLACK (Least slack): The job with smallest difference between due date and processing time is selected first. The final scheduled jobs are ordered satisfying:
$d_1 - p_1 \leq d_2 - p_2 \leq \ \ldots \leq d_n - p_n$.

Hodgson Algorithm: This algorithm gives an optimal schedule for the number of tardy jobs objective. The heuristic provides a schedule according to the following procedure,
Step 1: Order the activities using EDD heuristic.
Step 2: If there are no tardy jobs, stop; this is the optimal solution.
Step 3: Find the first tardy job, say *k*, in the sequence.
Step 4: Move the single job *j* ($1 \leq j \leq k$) with the longest processing time to the end of the sequence.
Step 5: Check the completion times and return to step 2.

4. Multirecombination of random and seed immigrants with the stud

Multiple Crossovers per Couple (MCPC) [6,7] and Multiple Crossovers on Multiple Parents (MCMP) [8] are multirecombination methods, which improve EAs performance by reinforcing and balancing exploration and exploitation in the search process. In particular, MCMP is an extension of MCPC where the multiparent approach proposed by Eiben [3-5] is introduced. Results obtained in diverse single and multiobjective optimization problems indicated that the searching space is

efficiently exploited by multiple applications of crossovers and efficiently explored by the greater number of samples provided by the multiple parents.

A further extension of MCMP is known as MCMP-SRI [13,14]. This approach considers the mating of an evolved individual (the stud) with random immigrants. The process for creating offspring is performed as follows. From the old population the stud is selected by means of proportional selection and inserted in the mating pool. 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 operation and $2*(n_2-1)$ offspring are created. The best of these $2*(n_2-1)$ 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.

As EAs are blind search methods our new variant (MCMP-SRSI) [15], proposes to insert problem-specific knowledge by recombining potential solutions (individuals of the evolving population) with seeds, which are solutions provided by other heuristics specifically designed to solve the scheduling problems under study. In MCMP-SRSI, the process for creating offspring is similar to that of MCMP-SRI, except that the mating pool contains also seed immigrants. In this way, the evolutionary algorithm incorporates problem-specific knowledge supplied by the specific heuristic. Figure 1 displays these processes.

We worked with different indirect representations: processor dispatching priorities and task priority list (both are indirect-decode representations) and another based on permutations.

The results discussed in next section correspond to EAs that worked on permutation-based representation using the PMX crossover operator because with this combination of representation and operator we obtained the best results.

5. Experimental Tests and Results

As it is not usual to find published benchmarks for the scheduling problems we worked on, we built our own test suite with data (p_j, d_j, w_j) based on selected data corresponding to weighted tardiness problems taken from OR library [1,2]. For problems sizes of 40 and 100 jobs, respectively, there were selected twenty problems each one with the same identification number although they are not the same problem, that is to say that we have a problem numbered 1, with 40 jobs and another with 100 jobs, and so on. The numbers of the problems are not consecutive because each one of them was selected randomly from different groups, where the tardiness factor varies being harder for those with highest identification number.

These data were the input for dispatching rules, conventional heuristics and our proposed EAs. To evaluate the dispatching rules and the conventional heuristics we used PARSIFAL [12] a software package provided by Morton and Pentico, to solve different scheduling problems by mean of different heuristics.

The initial phase of the experiments consisted in establishing the best results from dispatching rules and conventional heuristics to use them as upper bounds for the scheduling objectives. Also, the best parameter values for the EAs were empirically derived after performing a set of previous experiments. In all the experiments, we used population size 15 and we ran de EAs for 200 generations. The values of the remaining parameters are the following: crossover probability 0.65, $n_1 = 18$, $n_2 = 20$, seed number = 1 (only for MCMP-SRSI). For each problem and algorithm studied we performed 30 runs.

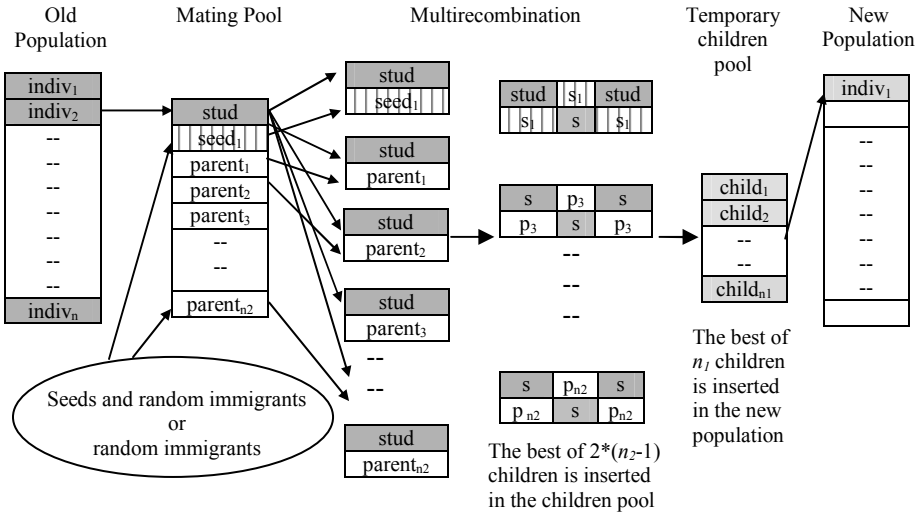

Figure 1. Stud and (random immigrants / seeds and random immigrants) multirecombination processes.

To compare the algorithms, the following relevant performance variables were chosen:

Ebest = ((best value - opt_val)/opt_val)*100
It is the percentile error of the best found individual when compared with the known or estimated (upper bound) optimum value opt_val. It gives a measure on how far the best individual is from that opt_val. When this value is negative, the op_val has been improved.
Mean Ebest (MEbest): It is the mean value of Ebest throughout all runs.
Best: It is the minimum objective value corresponding to some of the best found individuals throughout all runs.
Max Best: It is the maximum objective value corresponding to some of the best found individuals throughout all runs.
Mean Best (μBest): It is the mean objective value obtained from the best found individuals throughout all runs.
Gbest: It is the generation where the best individual was found.
Mean Gbest: It is the mean generation number where the best individual was found, throughout all runs.
Hit Ratio: Denotes the percentage of runs where the algorithm reaches the upper bound or improves it. Its value is 1 (a 100% of success) when the upper bound is reached or improved in every run.
Evals: Is the number of evaluations necessary to obtain the best found individual throughout all runs. The evaluation of an individual consists of calculating its fitness value.
Mean Evals (MEvals): Is the mean number of evaluations necessary to obtain the best found individual throughout all runs.
σBest: It is the standard deviation of the objective values corresponding to the best found individuals throughout all runs with respect to μBest.
(σ/μ) Best: This coefficient of variation it is calculated as the σBest and μBest ratio. It represents the deviation as a percentage of the value μBest. When this value is closer to zero higher is the robustness of the results obtained by the EA.

Several experiments were performed for 2 and 5 parallel equal machines scheduling systems for the objectives described before. The results obtained with the different multirecombined EAs implementations performed well for both scheduling systems. In this paper, due to space constraints, we only present the best results obtained corresponding to the 5 parallel equal machines scheduling problem with problems sizes of 40 and 100 jobs, respectively. Average values of 30 runs for the *MEbest* and *MEvals* performance variables, obtained under MCMP-SRI and MCMP-SRSI approaches, are showed in Figures 2 and 3, respectively.

Considering the precision of the results found by the EAs, it can be seen in Figure 2 that in general they performed very well with respect to the upper bounds. For the *maximum tardiness* objective, MCMP-SRI did not perform as well as we expect for the 100 jobs problems size. This can happens as a consequence of having a small population (only 15 individuals), besides the fact that the selected stud in each generation was not as good as needed to guide the search to more promising areas of the space of solutions. Taking into to account both objectives we worked on, MCMP-SRSI for all the instances reached or improved the values used as benchmarks.

In Figure 3, are shown the average number of evaluations done by the EAs. It can be observed that the most costly method was MCMP-SRI. Besides, the average number of evaluations done by MCMP-SRI increases for the 100 jobs problems size for the T_{max} and N_t objectives. The same occurs when considering the N_t objective for MCMP-SRSI. It is clear that augmenting the size of the scheduling problems produces an increase of the dimensionality in the search space of the algorithm. In general the average number of evaluations done by MCMP-SRSI is lower than MCMP-SRI due to the knowledge of the problem used to guide the search to promising areas of solutions. However MCMP-SRSI is still a costly method.

In Table 1 are presented the coefficients of variation calculated by the multirecombinated EAs for all the objectives and instances we worked on. Although not all the coefficients values are equal to 0.0 they are very close suggesting the algorithms' robustness with respect to the results that they found.

6. Conclusions

Multirecombined evolutionary algorithms have been successfully used to solve scheduling problems. In particular MCMP-SRSI, has demonstrated its ability on unrestricted identical parallel machine scheduling problems by improving the upper bounds calculated with different heuristics using PARSIFAL [12] and the values found by MCMP-SRI, for various problem data of different sizes taken from the OR-Library. MCMP-SRI also had a good performance for the problems of size 40 in both objectives and for the problems of size 100 when considering the N_t objective.

Comparing MCMP-SRSI with MCMP-SRI, the former finds good results with a lower cost (number of evaluations), due to the knowledge of the problem used to guide the search to promising areas of solutions. Considering the T_{max} objective the average numbers of evaluations done with MCMP-SRSI for both problems sizes (40 and 100 jobs) are similar. For the N_t objective the number of evaluations done with MCMP-SRSI for the 100 jobs size problems was higher than with the problems of 40 jobs size problems; this could be produced as a consequence on the inherent hardness of the scheduling problem, besides the fact that the seeds used for the T_{max} objective could be more suitable than the ones used for the N_t objective.

Figure 2. Average MEbest values found by the EAs

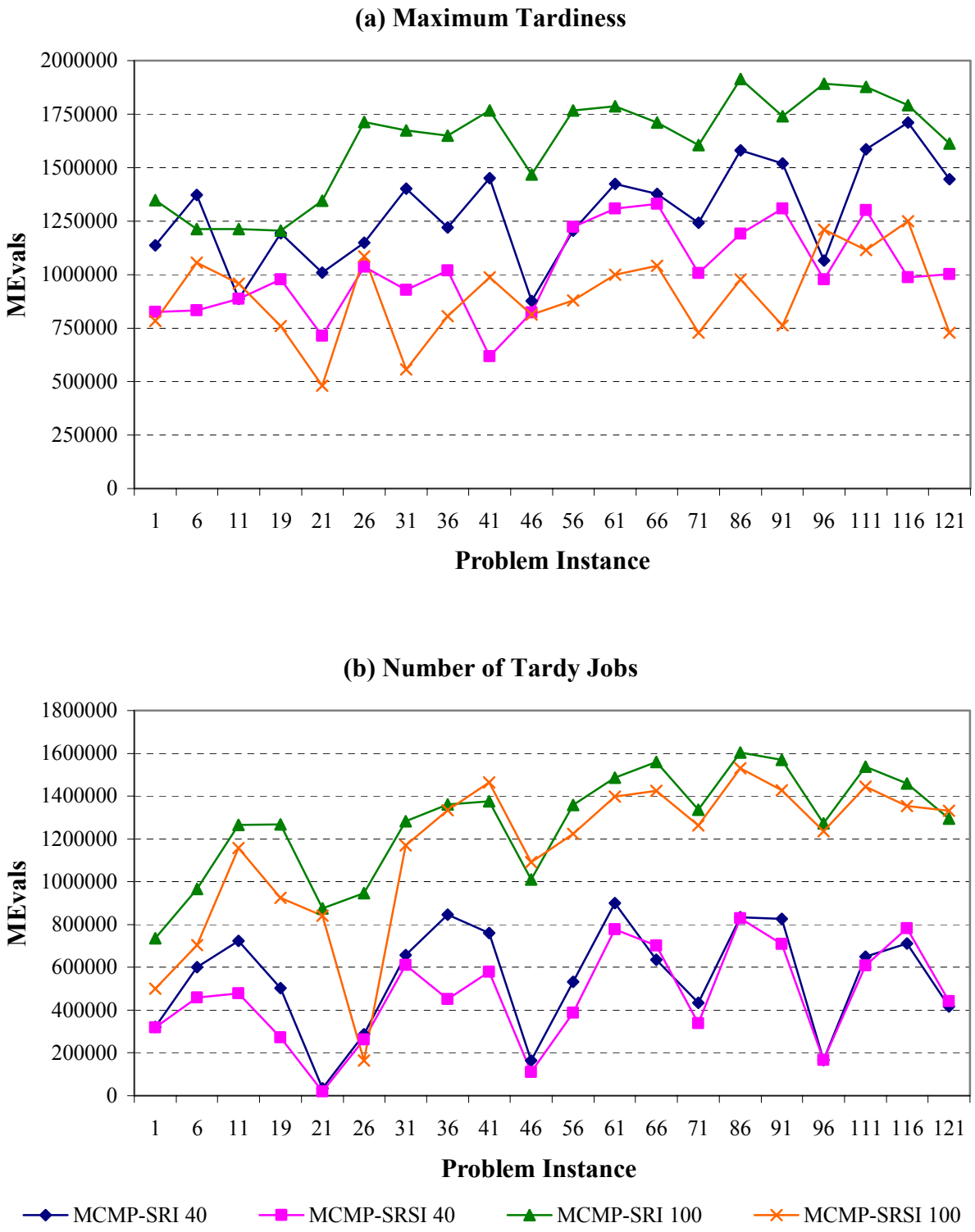

Figure 3. Average MEvals values found by the EAs

Table 1. (σ/μ) Best values found by the EAs

	40 Jobs				100 Jobs			
	MCMP-SRI		MCMP-SRSI		MCMP-SRI		MCMP-SRSI	
N°	T_{max}	N_t	T_{max}	N_t	T_{max}	N_t	T_{max}	N_t
1	0.05	0.09	0.02	0.08	0.04	0.07	0.00	0.08
6	0.02	0.07	0.00	0.05	0.01	0.04	0.01	0.03
11	0.01	0.04	0.00	0.03	0.01	0.03	0.01	0.02
19	0.01	0.02	0.00	0.02	0.00	0.03	0.00	0.02
21	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.01
26	0.17	0.12	0.10	0.26	0.22	0.22	0.12	0.13
31	0.03	0.06	0.02	0.07	0.05	0.05	0.00	0.05
36	0.02	0.04	0.01	0.04	0.04	0.04	0.01	0.04
41	0.03	0.04	0.00	0.04	0.03	0.03	0.00	0.02
46	0.01	0.00	0.01	0.00	0.01	0.01	0.00	0.02
56	0.09	0.13	0.02	0.11	0.12	0.09	0.02	0.09
61	0.06	0.09	0.02	0.06	0.06	0.05	0.01	0.05
66	0.03	0.05	0.01	0.03	0.05	0.03	0.01	0.05
71	0.02	0.01	0.01	0.02	0.02	0.02	0.00	0.01
86	0.06	0.08	0.03	0.06	0.08	0.05	0.00	0.06
91	0.03	0.03	0.01	0.05	0.04	0.03	0.00	0.03
96	0.01	0.00	0.01	0.01	0.03	0.02	0.00	0.01
111	0.08	0.04	0.02	0.05	0.07	0.05	0.00	0.05
116	0.06	0.05	0.02	0.05	0.04	0.02	0.01	0.02
121	0.03	0.01	0.01	0.01	0.04	0.02	0.00	0.02
AVG	**0.04**	**0.05**	**0.02**	**0.05**	**0.05**	**0.05**	**0.01**	**0.04**

Future work will be devoted to solve due date related problems in unrestricted identical parallel machine scheduling systems for larger number of jobs and to compare the performance of the different EAs implemented with others population-based stochastic search heuristics.

Acknowledgements

In memory of Dr. Raúl Gallard who was the inspirer of the present work, with all our respect and gratitude.

We acknowledge the Universidad Nacional de San Luis and the ANPCYT from which we receive continuous support.

References

[1] J.E. Beasley "Weighted Tardiness", OR Library, http://mscmga.ms.ic.ac.uk/

[2] H.A.J. Crauwels, C.N. Potts and L.N. Van Wassenhove, "Local Search Heuristics for the Single Machine Total Weighted Tardiness Scheduling Problem", Informs Journal on Computing 10, pp. 341-350, 1998.

[3] Eiben A.E., Raué P.E., and Ruttkay Z., "Genetic Algorithms with Multi-parent Recombination", Proceedings of the 3rd Conference on Parallel Problem Solving from Nature, number 866 in LNCS, pp. 78-87, Springer-Verlag, 1994.

[4] Eiben A.E., Van Kemenade C.H.M., and Kok J.N., "Orgy in the Computer: Multiparent Reproduction in Genetic Algorithms", Proceedings of the 3rd European Conference on Artificial Life, number 929 in LNAI, pp. 934-945, Springer-Verlag, 1995.
[5] Eiben A.E. and. Bäck T., "An Empirical Investigation of Multi-parent Recombination Operators in Evolution Strategies", Evolutionary Computation, 5(3):347-365, 1997.
[6] Esquivel S., Leiva A., Gallard R., "Multiple Crossover per Couple in Genetic Algorithms", Evolutionary Computation (ICEC'97), IEEE Publishing Co, 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 del International Symposium on Engineering of Intelligent Systems, University of La Laguna, España, Tenerife, Vol. 1, pp 235-241, ISBN 3-906454-12-6, February 1998.
[8] 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.
[9] Esquivel S., Ferrero S., Gallard R., Salto C., Alfonso H. and Schütz M., "Enhanced Evolutionary Algorithms for Single and Multiobjective Optimization in the Job Shop Scheduling Problem", Knowledge-Based Systems 15 (2002), pp. 13-25.
[10] Feltl H. and Raidl G., "An Improved Hybrid Genetic Algorithm for the Generalized Assignment Problem", In Proceedings of the 2004 ACM symposium on Applied computing, Nicosia, Cyprus, ECO, pp. 990 – 995, ISBN:1-58113-812-1, 2004.
[11] Min Liu and Cheng Wu, "Scheduling Algorithm based on Evolutionary Computing in Identical Parallel Machine Production Line", Robotics and Computer-Integrated Manufacturing 19 (2003), pp. 401-407.
[12] Morton T., Pentico D., "Heuristic Scheduling Systems", Wiley series in Engineering and technology management, John Wiley and Sons, INC, 1993.
[13] Pandolfi D., Vilanova G., De San Pedro M., Villagra A., Gallard R., "Multirecombining Studs and Immigrants in Evolutionary Algorithm to face Earliness-Tardiness Scheduling Problems", In Proceedings of the International Conference in Soft Computing, pp. 138, University of Paisley, Scotland, U.K., June 2001.
[14] Pandolfi D., De San Pedro M., Villagra A, Vilanova G., Gallard R., "Studs Mating Immigrants in Evolutionary Algorithm to Solve the Earliness-Tardiness Scheduling Problem", In Cybernetics and Systems of Taylor and Francis Journal, Vol.33 Nro. 4, pp. 391-400 (U.K.), June 2002.
[15] Pandolfi D., De San Pedro M., Villagra A., Vilanova G., Gallard R., "Multirecombining Random and Seed Immigrants in Evolutionary Algorithms to Solve W-T Scheduling Problems", In proceedings of CSITeA02, pp 133-138, Iguazu Falls, Brazil, June 2002.
[16] Pinedo M., "Scheduling: Theory, Algorithms and System", Prentice Hall, First edition, 1995.