INSERTING KNOWLEDGE IN MULTIRECOMBINED 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

Determining an optimal schedule to minimize the completion time of the last job abandoning the system (makespan) becomes a very difficult problem when there are more than two machines in the flow shop. Due both to its economical impact and complexity, different techniques to solve the Flow Shop Scheduling problem (FSSP) has been developed. Current trends addressed to multirecombination, involve distinct evolutionary computation approaches providing not a single but a set of acceptable alternative solutions, which are created by intensive exploitation of multiple solutions previously found.

Evolutionary algorithms perform their search based only in the relative fitness of each potential solution to the problem. On the other hand specialised heuristics are based on some specific features of the problem.

This work shows alternative ways to insert knowledge in the search by means of the inherent information carried by solutions coming from that specialised heuristic or gathered by the evolutionary process itself. The present paper compares the performance of multirecombined evolutionary algorithms with and without knowledge insertion and their influence in the crossover rate, the population size and the quality of results when applied to selected instances of the FSSP.

Keywords: flow shop scheduling, evolutionary computation, problem-specific-knowledge.

---

[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.

In the Flow Shop Scheduling Problem (FSSP), a number of operations must be done on every job and these operations have to be done in the same order on all jobs. Frequently, the main objective to be minimized here is the completion time of the last job to abandon the system, called the *makespan* [8,10]. This problem has been proved as NP-hard for even a very small number of resources. 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. [6, 8,10].

Attempting to provide a better balance between exploration and exploitation [7], 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) [2, 3, 4, 5] 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 set of alternative high-quality solutions are available. The multirecombined methods were applied to FSSP [1, 13] and contrasted on a series of suitable experiments against previous successful approaches of Tsujimura and Reeves [12]. A variant MCMP-SRI (multiple crossovers on multiple parents, stud and random immigrants) was recently proposed. Here, a stud (breeding individual) is selected from the old population, for recombination with a set of randomly generated parents. The members of this mating pool subsequently undergo multiple crossover operations.

For this work we designed alternative ways to insert knowledge in MCMP-SRI. In the first approach (MCMP-SRI-E1) the elitist individual always shares the mating pool with the random immigrants. In the second approach (MCMP-SRI-E2) the elitist individual is selected as the stud in every alternative generations. In the third approach (MCMP-SRSI-E) the elitist individual and the four seeds generated by GUPTA, PALMER, NEH, and CDS algorithms share the mating pool with the random immigrants. The following sections discuss these new proposals and the results obtained.

2. STUDS, RANDOM IMMIGRANTS AND SEEDS

Now we will describe in detail the schemes designed for multirecombination. In MCMP-SRI [9], 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. After each crossover operation offspring and parents compete for survival. The best of these $2*n_2$ individuals 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. Usually mutation is not applied because enough genetic diversity is afforded by the random immigrants. In an evolutionary process the elitist individual is the best solution found so far, and is retained by forcing its presence during subsequent generations until a better individual is found. This individual sums up the best characteristics of a solution gathered through the learning process of evolution. In MCMP-SRI-E1 the elitist individual is always present in the mating pool and shares it with the random immigrants. In this way the elitist individual continuously contributes with its genetic material each time the stud mates it. In MCMP-SRI-E2 the elitist individual is selected as the stud in every alternate generation. In other words, during an even generation the elitist individual is selected as the stud and mates every other random immigrant of the mating pool while in odd generations the stud is retrieved, as usually, as an individual of the evolving population by proportional selection. In (MCMP-SRSI-E) the elitist individual and the four seeds (GUPTA, PALMER, NEH, and CDS) share the mating pool with the random immigrants and all of them are recombined with the stud, which comes from the evolving population.

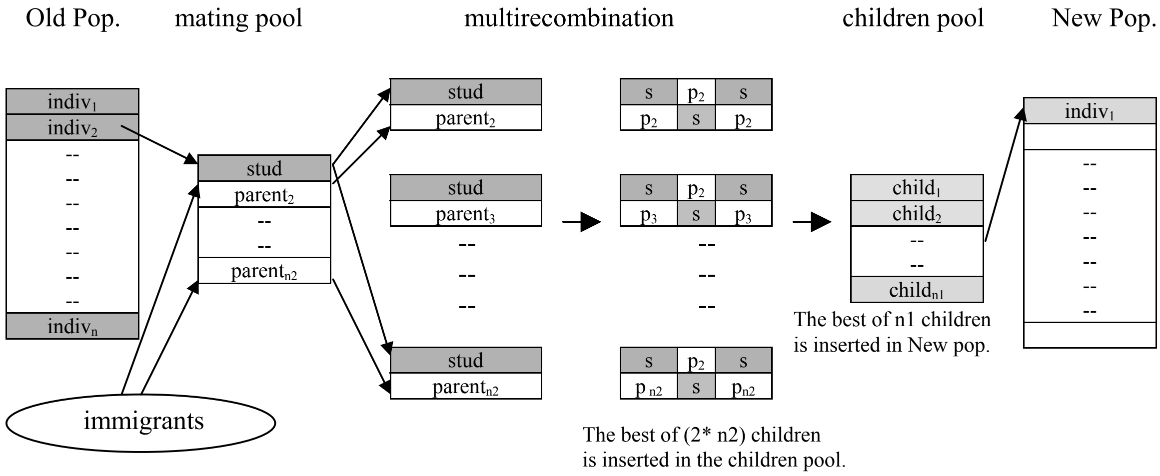

Fig. 1. The stud and random immigrants multirecombination process.

3. EXPERIMENTS AND RESULTS

To contrast the different approaches we selected four first instances extracted from Taillard´s FSSP benchmarks [11], for each of the following problem sizes: 20x5, 20x10, 20x20, 50x5, 50x10, 50x20. For each instance a series of 10 runs was performed. After a set of initial trials it was verified that better performance was achieved in the approaches inserting knowledge by always crossing over every pair of parents. Also, a non-null mutation probability value was necessary to avoid premature convergence induced by the elitist individual and the seeds. These trials allowed to reduce considerably the population size and consequently the computational effort. According with this, the following parameters setting was defined:

Approach	max_gen	pop_size	Pc	Pm	n1	n2
MCPC-SRI	100	100	0.65	0	6	8
MCPC-SRI-E1	100	10	1.0	0.3	10	14
MCPC-SRI-E2	100	10	1.0	0.3	10	14
MCPC-SRSI-E	100	10	1.0	0.3	10	14

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.

MEbest: It is the mean value of the error, over the total number of runs for each instance of a given problem size.

AvMEbest: It is the average value of the mean error, over the total number of runs, instances and problem sizes.

Figure 2, summarizes the results. Here *MEbest* and *Mean AvMEbest* are indicated for each algorithm applied to the selected instances of each problem size.

Problem size	MCMP-SRI	MCMP-SRI-E1	MCMP-SRI-E2	MCMP-SRSI-E
20x 5	4.225	1.299	**1.249**	1.855
20x 10	7.998	3.661	5.497	**3.650**
20x 20	6.290	**1.842**	4.944	2.855
50x 5	3.614	**1.192**	2.365	1.642
50x 10	12.227	9.696	8.986	**6.736**
50x 20	13.514	11.062	11.159	**8.134**
Avg	7.978	4.792	5.700	***4.145***

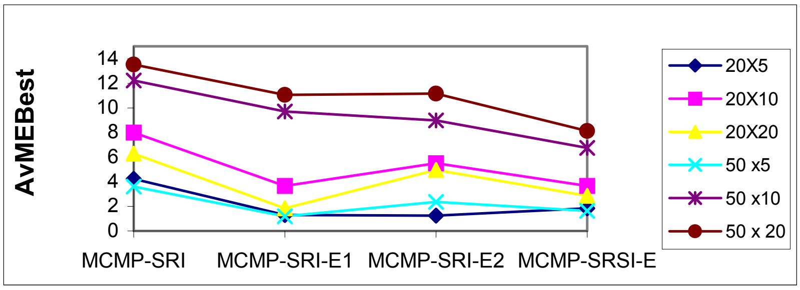

Fig. 2. MEbest and AvMEbest values for each method on different problem sizes for the FSSP

The comprehensive average results obtained from a large series of experiments on different problem sizes for FSSP shows that any of the versions inserting knowledge into the EA outperform the versions where this knowledge is not included. For different problem sizes the best performers are: MCMP-SRI-E2 in 20x5, MCMP-SRI-E1 in 20x20 and 50x5 and MCMP-SRSI-E in 20x10, 50x10 and 50x20. The overall best performer resulted MCMP-SRSI with an average mean error of 4.1%.

4. CONCLUSIONS

Previous multirecombined evolutionary approaches for solving the FSSP attempted to improve the algorithms performance by balancing exploration and exploitation in the searching space. As that, they merely are blind search algorithms, which only make use of the relative fitness of the solutions, but completely ignore the nature of the problem. The issue here is how to introduce knowledge, which is specific to the problem?. 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, the answer is to provide information which is gathered by the evolution process itself and resides in the elitist individual, or to import this knowledge from solutions that come out from heuristics specifically designed for the problem under consideration. Both kinds of knowledge-based intermediate solutions contain some of the features, which are present in the best (optimal or quasi-optimal) solution at the end of the evolutionary process.

The mere presence in the population of the best individual found so far does not guarantee that it will be selected for mating. Consequently, in the first two variants we reinforced the contribution of the elitist individual. In MCMP-SRI-E1 we compelled the presence of the elitist individual in the mating pool of multiple parents. In this way, each time it is recombined with the stud, we ensure a contribution of this individual in the offspring that eventually go to the next generation. In the second variant MCMP-SRI-E2 we forced the elitist individual to act as the stud every alternate generation, making stronger its contribution. As a consequence the average mean error increased. This should be reviewed. Finally in MCMP-SRSI-E we combined self and foreign knowledge by allocating in the mating pool the elitist individual together with the seeds. This latter approach resulted to be the best performer in average.

As all approaches inserting knowledge outperform the previous approach, with lesser computational effort, further work will be dedicated to find alternative ways to help, through knowledge insertion, the evolutionary search for different scheduling problems.

5. ACKNOWLEDGEMENTS

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

6. 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 International ICSC Symposium of Intelligent Systems EIS 2000, June 27-30 , 2000, University of Paisley, Scotland U.K.

2. 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

3. 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.

4. 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.

5. 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.

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

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

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

9. Pandolfi D., Vilanova G., 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.

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

11. 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).

12. 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.

13. Vilanova G., Pandolfi D., De San Pedro M., Villagra A., Bain M, Gallard R. Deterministic parameter control in multirecombinated evolutionary algorithms for the flow shop scheduling problem . Proceedings of the SCI 2001 Orlando. USA, June 2001.