INCEST PREVENTION AND MULTIRECOMBINATED EVOLUTIONARY ALGORITHMS FOR THE JOB SHOP SCHEDULING PROBLEM

Salto Carolina, Stark Natalia, Alfonso Hugo
Proyecto UNLPAM-09/F015[1]
Departamento de Informática - Facultad de Ingeniería
Universidad Nacional de La Pampa
Calle 110 esq. 9
(6360) General Pico – La Pampa – Rep. Argentina
e-mail: { saltoc, nstark, alfonsoh }@ing.unlpam.edu.ar
Phone: +54 (2302) 422780/422372, Ext. 6302

Gallard Raúl
Proyecto UNSL-338403[2]
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
(5700) - San Luis -Argentina
e-mail: rgallard@unsl.edu.ar
Phone: +54 (2652) 420823
Fax : +54 (2652) 430224

SUMMARY

Evolutionary algorithms (EAs) have been successfully applied to scheduling problems. Current improvements towards convergence issues in EAs include incest prevention and multiplicity features.
A multiplicity feature allows multiple recombination on multiple parents [7, 8, 9, 10]. The method was successfully applied to multimodal optimization problems. As a consequence of this approach it was detected that all individuals of the final population are much more centred on the optimum. This is an important issue when the application requires provision of multiple alternative near-optimal solutions confronting system dynamics as in production planning. The multiplicity feature here discussed is related to new proposed multi-recombination methods:

- ✓ MCPC: *Multiple Crossovers per Couple,* which reinforces the exploitation of features of previously found (good) solutions.
- ✓ MCMP: *Multiple Crossovers on Multiple Parents*, which provides a balance in exploitation and exploration because the searching space is efficiently exploited (by the multiple application of crossovers) and explored (by a greater number of samples provided by multiple parents). MCMP provides a means to exploit good features of more than two parents selected according to their fitness by repeatedly applying one of the scanning crossover (SX) variants [5]: a number n_1 of crossovers is applied on a number n_2 of selected parents. From the n_2 produced offspring a number n_3 of them are selected, according to some criterion, to be inserted in the next generation. This is the method used in all our experiments.

The idea of *incest prevention* was initially proposed by Eshelman and Shaffer [6] and showed its benefits to avoid premature convergence. The method avoided mating of pairs showing similarities based on the parents' hamming distance. Incest prevention was extended in [1] by maintaining information about ancestors within the chromosome and modifying the selection for reproduction in order to prevent mating of individuals belonging to the same "family", for a predefined number of generations. This novel approach was also tested on a set of multimodal functions.
In scheduling the main difficulty encountered is to specify an appropriate representation of feasible schedules (the solution of JSSP) [3]. The representations can be direct or indirect [2].
Two different indirect representations are used. They are the *decoders* and the *priority rule based* representation. Both are domain-independent representations, then they do not contain auxiliary information of the particular scheduling problem.

Using decoders is another way to face a problem involving permutations. A chromosome is an *n*-vector where the i^{th} component is an integer in the range 1..(n-i+1). The chromosome is interpreted as a strategy to extract items from an ordered list L and build a permutation.

---

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

The use of decoders for permutation problems allows the direct implementation of conventional crossovers avoiding the need of penalties or repair actions. This is strongly true in the multirecombination approaches where the SX methods as initially defined are not applicable for permutation representations.
In [17] a multirecombinative approach MCMP and its further combination with incest prevention MCMPIP was introduced, solving the JSSP for a set of instances [15] of distinct complexity. After long series of experiments, better results were obtained when the number n_1 of crossovers was augmented for a given number n_2 of parents. In many cases, the final population was centred on the fittest individual, providing a set of alternative schedules, which can help to deal with system dynamics.

With a priority-rule-based representation [4, 12] a chromosome is encoded as a sequence of dispatching rules for job assignment and a schedule is built with a priority dispatching heuristic based on the sequence of dispatching rules. Evolutionary algorithms are used to evolve those chromosomes improving the sequences of dispatching rules. Priority dispatching rules are frequently applied heuristics for solving scheduling problems due to their ease of implementation and low time complexity. Giffler and Thompson´s algorithms can be considered as the basis of priority rule based heuristics [11]. The main problem is to determine an effective priority rule.
In [19] a hybrid multirecombinative evolutionary algorithm (MCMP-PRB) to solve the JSSP was introduced. It used both multiple crossovers on multiple parents and priority dispatching rules. By means of the rule based representation MCMP could be applied creating valid offspring after each recombination operation. Repair algorithms or penalty functions were no needed. The approach was tested on a set of instances of varied complexity [15]. After this preliminary experiments we conclude that MCMP-PRB provided optimal solutions on most of them. It is remarkable also the fact that the composition of the final population guarantees the supply of many near optimal solutions that can help facing changes in system dynamics.
A comparison of the two previous evolutionary approaches to solve the JSSP appeared in [16]. MCMP-PBR reached the optimum for any (n_1, n_2) combination for *la06*, *la01*, *la12* and *la15* instances. When instance complexity increased it became harder for both algorithms to find the optimum and this problem is stronger under MCMP-Dec (MCMP option with decoder representation) where a tendency to stagnate the search is detected. In general, MCMP-PBR performs better than MCMP-Dec.
In a direct representation, like job based representation [14], a schedule is encoded into a chromosome and evolutionary algorithms are used to evolve those chromosomes to determine a better schedule. Using this kind of representation is necessary to consider special crossover operators. The conventional ones (one-point or multipoint crossover) generally might produce illegal offspring in the sense that some jobs may be missed while some others may be duplicated in the offspring. There are some specific crossover operators designed to apply to a permutation such as: partial-mapped crossover (PMX), order crossover (OX), cycle crossover (CX), order-based crossover, and so on.
Due to the use of permutations in the chromosome representation, the traditional SX methods were not applied in multirecombination, because they can produce infeasible offspring. Besides, the existence of special scanning crossover to work with permutations, such *adjacency based crossover* (ABC) [5], in [18] another way of multi-recombination is considered. In this case the PMX operator replaces the scanning method. From the n_2 parents selected, the best one is called the *stud* (the breeding individual) for that group. Two crossover points are determined in a random way, the stud is combined with the rest of the parents selected using PMX. From that multi-recombination, $2*(n_2 -1)$ offspring are obtained, but only the best one survive. The preceding crossover process is repeated n_1 times. Finally n_1 offsprings are obtained and the best one is selected to pass to the next generation.
An evolutionary algorithm using MCMP with PMX creating only the offspring that preserves the stud substring was contrasted to one using the conventional single crossover per each couple of parents (SCPC) with conventional PMX creating two offspring per mating action. These approaches were used to solve the JSSP for a set of instances of distinct complexity. After long series of experiments we can conclude that the multirecombinative approach (MCMP) outperforms the conventional one (SCPC) in each performance variable. In general, better results are obtained when 4 crossovers are applied than when 1 crossover is applied. In many cases the final population is centred on the fittest individual, providing a set of alternative schedules, which can help to deal with system dynamics.

CONCLUSIONS AND FUTURE WORK

This contribution presents a multirecombinative approach MCMP and its further combination with incest prevention, to solve the JSSP for a set of instances of distinct complexity using diverse representations.
All the cases presented in this report, dealing with multiplicity features, outperform a simple evolutionary algorithm. Considering that promising results reached incorporating the multiplicity feature encourage us to deep investigation in this area.
Future work will be oriented to introduce different representations into an evolutionary algorithm gathered for the JSS problems, mainly those that belong to the category of direct representations.

To improve results in complex instances, further works include self adaptation of parameters such as (n_1,n_2) associations, population size and probabilities of crossover and mutation.

REFERENCES

[1] Alfonso H., Cesan P., Fernandez N., Minetti G., Salto C., Velazco L., Gallard R. –*Improving Evolutionary Algorithms Performance by Extending Incest Prevention*- Proceedings del 4to Congreso Argentino de Ciencias de la Computación (CACiC'98), pp 323-334. Abstracts del Congreso, pp 69, Universidad Nacional del Comahue, October 1998.

[2] Bagchi, S., Uckum, S., Miyabe, Y., and Kawamura, K., *Exploring Problem-Specific Recombination Operators for Job Shop Scheduling*, in Proceedings of the fourth International Conference on Genetic Algorithms, San Diego, Morgan Kaufmann Publishers, 1991.

[3] Bruns, R. Scheduling, in Th. Bäck, D. B. Fogel, and Z. Michalewicz, (editors), *Handbook of Evolutionary Computation* , chapter F1.5, pages F1.5:1-F1.5:9. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.

[4] Dorndorf, U., and Pesch, E., *Evolution based learning in a job shop scheduling environment*, Computers and operations Research, vol. 22, pp. 25-40, 1995.

[5] 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.

[6] 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.

[7] Esquivel S., Leiva A., Gallard R., Multiple Crossover per Couple in Genetic Algorithms, in *Proceedings of the 4th IEEE International Conference on Evolutionary Computation* (ICEC'97), pages 103-106, Indianapolis, USA, April 1997.

[8] Esquivel S., Leiva A., Gallard R., Couple Fitness Based Selection with Multiple Crossover per Couple in Genetic Algorithms, in *Proceedings of the International Symposium on Engineering of Intelligent Systems* (EIS´98), La Laguna, Tenerife, Spain Vol. 1, pages 235-241, ed. E.Alpaydin. Publishesd by ICSC Academic Press, Canada/Switzerland, February 1998.

[9] Esquivel, S., Leiva, A. And Gallard, R., Multiple Crossovers between Multiple Parents to Improve Search in Evolutionary Algorithms, en *Proceedings of the 1999 Congress on Evolutionary Computation (IEEE)*, vol. 2, pages 1589-1594, Washington DC., 1999.

[10] Esquivel, S., Leiva, A. and Gallard, R., Multiplicity in Genetic Algorithms to Face Multicriteria Optimization, en *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE Service Center, pages 85-90, Washington, D.C., 1999.

[11] Garey, M., and Johnson, D., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

[12] Gen, M. y Cheng, R., *Genetic Algorithms and Engineering Design*, Wiley-Interscience Publication John Wiley & Sons, Inc, 1997.

[13] Goldberg, D. y Lingle, R., Alleles, Loci and the Traveling Salesman Problem, en Grefenstette, J. (editor), *Proceedings of the First International Conference on Genetic Algorithms,* Lawrence Erlbaum Associates, pages 154-159, Hillsdale, NJ, 1985.

[14] Holsapple, C., Jacob, V., Pakath, R., and Zaveri, J., A Genetics-Based Hybrid Scheduler for Generating Static Schedules in Flexible Manufacturing Contexts, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, pages 953-971, 1993.

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

[16] Minetti G., Salto, C., Alfonso, H., and Gallard, R., A Comparison of two Multirecombinated Evolutionary Algorithms for the Job Shop Scheduling Problem, in *Proceedings VI Congreso Argentino de Ciencias de la Computación*, CACIC 2000 Universidad Nacional de la Patagonia San Juan Bosco, 2000.

[17] Salto, C., Alfonso, H., and Gallard, R., Multiplicity and Incest Prevention in Evolutionary Algorithms to Deal with de Job Shop Problem, in *Proceedings of the Second ICSC Symposium on Engineering of Intelligent Systems*, University of Paisley, Scotland, pag, 451-457, 2000.

[18] Salto, C., Alfonso, H., and Gallard, R., Breeding in multirecombinated evolutionary algorithms with permutation based representation for the job shop scheduling problem, submitted to SOCO 2001. Scotland U.K.

[19] Salto, C., Minetti G., Alfonso, H., and Gallard, R., A Hybrid Evolutionary Algorithm: Multirecombination with Priority Rule Based Representation for the Job Shop Scheduling Problem, in *Proceedings VI Congreso Argentino de Ciencias de la Computación*, CACIC 2000, Universidad Nacional de la Patagonia San Juan Bosco, 2000.