USING DIFFERENT CHROMOSOME REPRESENTATIONS FOR THE JSS PROBLEM

Salto Carolina, 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,alfonsoh }@ing.unlpam.edu.ar
Phone: (02302)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

Nearly all practical scheduling problems can be described in terms of the job-shop scheduling problem, in which *n* jobs are to be processed by *m* machines. Each job will have a set of constraints on the order in which machines can be used; moreover the processing time on each machine is specified for each job. The job-shop scheduling problem consists in finding a sequence of jobs on each machine in order to minimise a given objective function. In the case here considered, it is the minimization of the makespan.

A crucial aspect of the research on the job-shop scheduling problem [4, 14, 16] is the method used to encode schedules. Evolutionary algorithms, when applied to scheduling, visualises schedules (the solution of a JSSP) as individuals of a population.

A schedule could simply be a list specifying the order of the operations to be performed by each machine. But simple crossover applied to such strings would nearly always result in illegal offspring with some operations missing, others represented twice, etc. Hence more sophisticated crossover operations are needed. Many other GA applications to scheduling problems were done by others, such as, [10], [6] and [5].

From the representation perspective many evolutionary computation approaches to the general scheduling problem exists. According to solution representation these methods can be roughly categorised as *indirect* and *direct representation* ([2, 1]).

In direct representation [3], the schedule produced is an individual of the evolving population. Decoder procedure is not necessary. The only method that performs the search is the evolutionary algorithm because the represented information comprises the whole search space. The problem arising with this kind of representation is to design specific genetic operators.

In the case of indirect representation of solutions a transition from chromosome representation to a legal production schedule has to be performed. A schedule builder, who guarantees the feasibility of a solution, carries out this transformation and its work depends on the amount of information included in the representation. In this case the algorithm works on a population of encoded solutions.

Until this moment, we have been concentrated in the use of four different representations: job based representation (JBR) [12], operation based representation (OBR) [9], priority dispatching rules (PDR) [5] and decoders (Dec). The first two are direct representations while the later ones are indirect representation.

---

[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).

If those indirect representations are considered, conventional genetic operators can be applied creating valid offspring after each recombination operation. Repair algorithms or penalty functions are no needed, as under other evolutionary approaches. For example, for PDR and JBR the conventional one-point crossover can be applied. Instead with JBR the use of PMX [11] is advisable, because chromosomes are permutations; generally, permutations will yield to invalid offspring by using two point or multipoint crossover in the sense that some jobs may be missed while some other jobs may be duplicated in the offspring. A repairing procedure is embedded in this approach to resolve the illegitimacy of offspring. Something similar occurs with OBR, when a partial schedule exchange crossover (PSX) [9, 8] is used.

For mutation big creep and swapping is applied when it corresponds.

In [1] a comparative study of different crossovers operators suitable for Dec and JBR were analysed. PMX, CX and OX were applied to JBR and were compared. PMX works well for this kind of problem. One-point, two-point and uniform crossovers were used for JBR, where the first one outperforms the remaining representations.

When priority-rule-based representation is used, the well-known Giffler and Thompson algorithm [7, 15] was incorporated into the evolutionary algorithm. Here, the evolutionary algorithm is used to evolve a sequence of priority dispatching rules and the Giffler and Thompson algorithm is used to deduce a schedule from the encoding of priority dispatching rules.

CONCLUSIONS AND FUTURE WORK

Some evolutionary algorithms using different chromosome representations and associated operators to solve selected instances of the JSSP [13] were implemented. From the experiments we can say that incorporating more domain knowledge inside the chromosome enhance the behaviour of the algorithm when good solutions are looked for. From all the considered representations, OBR is the best one.

When good results are waited for, an order on goodness of representations can be established. OBR is in the first place, followed by PDR, JBR and finally Dec.

Future work will be oriented to consider another representations available in the literature for the JSS problems, besides testing the behaviour with another genetic operators.

REFERENCES

1. Alfonso, H., Salto, C., and Gallard, R., Análisis Comparativo de la Aplicación de Distintos Operadores Genéticos al Problema de Job Shop Scheduling, in *Anales del Workshop de Investigadores de Ciencias de la Computación*, WICC 2000, Universidad Nacional de La Plata, 2000.
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., Direct chromosome representation and advanced genetic operators for production scheduling. In Forrest ICGA93 , pp. 352-359.
4. Davis, L., *Job Shop Scheduling with Genetic Algorithms*, in Proceedings of the First International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, Hillsdale, pp. 136-140, 1985
5. Dorndorf, U., and Pesch, E., *Evolution based learning in a job shop scheduling environment*, Computers and operations Research, vol. 22, pp. 25-40, 1995.
6. Fox, B. and McMahon, M.. Genetic operators for sequencing problems. In G. Rawlins, editor, *Foundations of Genetic Algorithms*. Morgan Kaufman, 1991.
7. Giffler, B., and Thompson, G., *Algorithms for solving production scheduling problems*, Operations Research, vol. 8, no. 4, pp. 487-503, 1960.
8. Gen, M. y Cheng, R., *Genetic Algorithms and Engineering Design*, Wiley-Interscience Publication John Wiley & Sons, Inc, 1997.

9. Gen, M., Tsujimura, Y., y Kubota, E., Solving Job-Shop Scheduling Problem Using Genetic Algorithms, en Gen, M., y Kobayashi, T. (editores), *Proceedings of the 16th International Conference on Computers and Industrial Engineering*, Ashikaga, Japan, pag. 576-579, 1994.
10. Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
11. 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.
12. Holsapple, C., Jacob, V., Pakath, R., y Zaveri, J., A Genetics-Based Hybrid Scheduler for Generating Static Schedules in Flexible Manufacturing Contexts, en *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, pag. 953-971, 1993.
13. 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.
14. Nakano, R., and Yamada, T., *Conventional genetic algorithms for job-shop problems*, in Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Sap Mateo, CA, pp. 477-479, 1991.
15. Storer, R., Wy, S., and Vaccari, R., *New search spaces for sequencing problems with application to job shop scheduling,* Management Science, vol. 38, no. 10, pp. 1495-1510, 1992.
16. Whitley, T. Starkweather, and D. Fuquay, *Scheduling problems and travelling salesman: The genetic edge recombination operator.* In J.D. Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, pages 133-140, Morgan Kaufman, 1989.