Knowledge Insertion: An Efficient Approach to Reduce Effort in Simple Genetic Algorithms for Unrestricted Parallel Equal Machines Scheduling

[Extended Abstract]

Edgardo Ferretti
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
San Luis, Argentina
ferretti@unsl.edu.ar

Susana Esquivel
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
San Luis, Argentina
esquivel@unsl.edu.ar

ABSTRACT

Simple Genetic Algorithms (SGAs) are blind search algorithms which only make use of the relative fitness of solutions and completely ignore the nature of the problem. The SGAs have been used to solve different scheduling problems but in large search spaces, a considerable number of evaluations are required to obtain solutions nearer to the optimum (known or estimated). Our purpose was to try to reduce the number of evaluations by introducing problem specific knowledge through the insertion of good seeds (solutions) obtained with other conventional heuristics. This work shows how the knowledge insertion in a SGA, reduces the cost in solving due-date based problems in parallel machines scheduling systems.

Categories and Subject Descriptors

I.2.8 [**ARTIFICIAL INTELLIGENCE**]: Problem Solving, Control Methods, and Search—*Heuristic methods, Scheduling*

General Terms

Algorithms

Keywords

Artificial Intelligence, Genetic Algorithms, Scheduling.

1. SCHEDULING PROBLEMS

The problem we are facing [4] can be stated as follows: n jobs are processed without interruption on some of the m equal machines belonging to the system (P_m); 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 $T_j = \max\{C_j\text{-}d_j,0\}$ 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:

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

$$Average\ Tardiness: \quad T_{avg} = \frac{1}{n}\sum_{j=1}^{n} T_j$$

$$Weighted\ Tardiness: \quad T_{wt} = \sum_{j=1}^{n} w_j T_j$$

These problems have received considerable attention by different researchers. For most of them, for many years their computational complexity remained as an open research topic until established as NP-Hard [3].

2. THE EVOLUTIONARY APPROACH

Genetic Algorithms (GAs) have been successfully applied to solve scheduling problems. GAs are blind search algorithms which only make use of the relative fitness of solutions and completely ignore the nature of the problem. In difficult problems with large search spaces, a considerable number of evaluations are required by the GAs to obtain near-optimal solutions. In this paper we present a variant of a SGA, SGA-SE, that considers the inclusion of 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 SGA-SE, the evolutionary process is similar to that of SGA, except that the individuals of the population also mates with seeds. In this way, the SGA incorporates problem-specific knowledge supplied by the specific heuristics.

3. EXPERIMENTAL DESIGN

As it is not usual to find published benchmarks for the scheduling problems we worked on, we built our own test

GECCO'05, June 25–29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

suite with data (p_j, d_j, w_j) based on selected data corresponding to weighted tardiness problems taken from the OR library [1, 2]. For problems sizes of 40 and 100 jobs, respectively, were selected twenty problems. These data were the input for dispatching rules, conventional heuristics and the implemented GAs with and without knowledge insertion. To evaluate the dispatching rules and the conventional heuristics we used PARSIFAL [3] a software package provided by Morton and Pentico to solve different scheduling problems by means 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 GAs were empirically derived after performing a set of previous experiments. In all the experiments, we used population size 15 and we ran de GAs up to a maximum number of 5000 generations. The values of the remaining parameters are the following: crossover probability 0.65, mutation probability 0.05, and seeds number = 1 (only for SGA-SE). It is important to note that the seed provided to the GA is the one corresponding to the best value used as benchmark. For each problem and algorithm studied we performed 30 runs. 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 Evals (MEvals): Is the mean number of evaluations necessary to obtain the best found individual throughout all runs.

Several experiments were performed for 2 (P_2) and 5 (P_5) parallel equal machines scheduling systems for the objectives described before.

4. RESULTS

In this section a brief overview of the results is presented. When we use the term *precision* we mean how close is the value found by the GA with respect to the benchmark. If higher is the precision of the results better is the improvement when compared with the benchmark.

Considering the precision of the results, some general comments could be made:

- The SGA did not reach most of the benchmarks for the T_{max} objective neither in P_2 nor in P_5, whereas SGA-SE improved the upper bounds of all the instances with 40 and 100 jobs, respectively.
- The SGA enhanced its performance for the T_{avg} and T_{wt} objectives, improving the upper bounds of the conventional heuristics and getting also best values than SGA-SE on several instances in P_2 and in P_5, with both instances sizes.

An important point to consider to evaluate the performance of the GAs, is the cost of the search process they accomplish. In Figure 1, we present a plot of the MEvals values obtained by the GAs, that is a representative example of what was obtained with all the objectives in P_2 and in P_5, with both instances sizes. The MEvals values of SGA are always higher than those obtained by SGA-SE, therefore, it could be stated that SGA is a more costly method than SGA-SE.

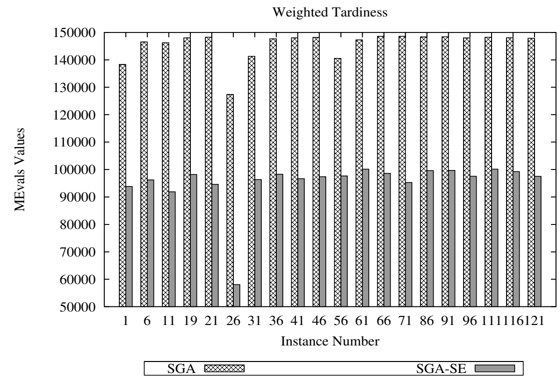

Figure 1: Weighted Tardiness MEvals values obtained by the GAs in P_5 with 100 jobs instances size.

5. CONCLUSIONS

It is known that in some cases seeding individuals in the population is equivalent of running for a few more generation the GA without seeding individuals. However, in our studies we have concluded that this is true for small and medium scheduling problem sizes, and also depends on the hardness of problem instance and the objective function. Even if SGA successfully solved some scheduling problems, in particular SGA-SE, have demonstrated its ability on unrestricted parallel machine due-date based scheduling problems, by improving the upper bounds calculated with different conventional heuristics for various problem data taken from the OR-Library. When comparing SGA-SE with SGA from a benefit/cost point of view, 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 space. In this way, using seeds that mates individuals of the population instead of running the GA for more generations has the advantage of speeding the convergence of the GA towards good solutions at a lower cost, and also helps finding solutions than in other cases SGA did not find.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] J. Beasley. Or library. http://people.brunel.ac.uk/~mastjjb/jeb/info.html. Scheduling: Weighted Tardiness.

[2] H. A. J. Crauwels, C. N. Potts, and L. N. VanWassenhove. Local search heuristics for the single machine total weighted tardiness scheduling problem. *INFORMS J. on Computing*, 10(3):341–350, 1998.

[3] T. Morton and D. Pentico. *Heuristic Scheduling Systems*. John Wiley and Sons, New York, 1993.

[4] M. Pinedo. *Scheduling: Theory, Algorithms and System*. Prentice Hall, 1995.