OPTIMIZATION OF TARDINESS RELATED OBJECTIVES IN SINGLE MACHINE ENVIRONMENTS VIA MULTIRECOMBINED EVOLUTIONARY ALGORITHMS

De San Pedro M., Villagra A., Lasso M., Pandolfi D., Vilanova G., Díaz de Vivar M.
Proyecto UNPA-29/B032[1]
División Tecnología
Unidad Académica Caleta Olivia
Universidad Nacional de La Patagonia Austral
Ruta 3 Acceso Norte s/n
(9011) Caleta Olivia – Santa Cruz - Argentina
e-mail: {mlasso,dpandolfi,edesanpedro,avillagra}@uaco.unpa.edu.ar; gvilanov@satlink.com}
Phone/Fax : +54 0297 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 -Argentina
e-mail: rgallard@unsl.edu.ar
Phone: +54 2652 420823
Fax : +54 2652 430224

Abstract

Tardiness related objectives are of utmost importance in production systems when client satisfaction is a main goal of a company. These objectives measure the system response to the client requirements and rate manager´s performance

In scheduling problems with diverse single or multiple objectives and environments Evolutionary algorithms (EAs) were successfully applied. Latest improvements in EAs have been developed by means of multirecombination, a method, which allows multiple exchange of genetic material between individuals of the mating pool. These individuals can be provided by the current population or by an external source. The performance of the algorithm depends o the number of individuals in the mating pool and their mating frequency.

MCMP-SRI and *MCMP-SRSI* are multirecombined evolutionary approaches using the concept of the stud (a breeding individual), random immigrants and/or seeds, to avoid premature convergence and adding problem-specific- knowledge. Here, both methods applied to tardiness related problems in single machine environmen are discussed and contrasted against conventional heuristics.

[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

Complex machine environments such as parallel machines are based in results provided by single-machine scheduling problems. Many single machine problems are NP-hard. This is the case for the, *Weighted Tardiness, Average Tardiness* and *Weighted Number of Tardy Jobs* problems [3, 8,11]. Branch and Bound, Beam Search, Dynamic Programming and other implicit enumeration based methods, which guarantee exact solutions, are prohibitively time consuming even with only 20 jobs. 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 of them behave better than others, but only in special cases they guarantee convergence to the optimal solution. This is also the case in evolutionary algorithms (EAs), but they have been successfully applied to solve scheduling problems [13,14] providing better solutions. Current trends in evolutionary algorithms make use of multiple parents [4, 5] and multirecombinative approaches [6, 7]. The latter we called, *multiple-crossovers-on-multiple-parents* (MCMP). Opposite to traditional EAs, which applies crossover once on a pair of parents, this mrthod applies n_1 crossover operations on a set of n_2 parents. By means of an intensive exploitation of good characteristics in a wider sample of the problem space the balance between these central features of an EA is improved. A variant called MCMP-SRI [9], recombines a breeding individual (stud) by repeatedly mating individuals that randomly immigrates to a mating pool. Under this approach the random immigrants incorporate exploration (making unnecessary the use of mutation operations, in some cases) and the multi-mating operation with the stud incorporates exploitation to the search process. In this work MCMP-SRSI, a latest variant, which considers the inclusion of a stud-breeding individual in a pool of random and seed-immigrant parents, is presented. Here, the seeds generated by conventional heuristics, continuously present in the mating pool, introduce the problem-specific-knowledge. Both evolutionary algorithms are contrasted against conventional heuristics, some of them also known as dispatching rules, which determine a priority order to build a schedule. Next sections describe tardiness-related objectives and the application of the new MCMP variants to solve the problems in a single machine environment.

2. Single machine tardiness-related scheduling problems

Single-machine tardiness-related problems [11] can be stated as follows: n jobs are to be processed without interruption on a single machine that 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 pj on the machine, has a positive weight wj, and a due date dj 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 tardiness-related problems that we consider here consists in finding a processing order of the jobs which minimizes one of the following objectives:

Weighted Tardiness: $\sum_{j=1}^{n} w_j T_j$, Average Tardiness: $\frac{1}{n}\sum_{j=1}^{n} T_j$

Weighted Number of Tardy Jobs: $$N_{wt} = \sum_{j=1}^{n} w_j \delta(T_j), \text{ with } \quad \begin{aligned} &\delta(T_j) = 1 \text{ if } T_j > 0 \\ &\delta(T_j) = 0 \text{ otherwise} \end{aligned}$$

Even with their simple formulation, these models lead to an optimization problem that is NP-Hard [11].

3. Typical approaches to tardiness-related scheduling problems

Dispatching heuristics are methods that allow deciding which job should be inserted into the system. To do this, a rule assigns an index to every job and the one with the highest priority is selected. There are different heuristics [8] for the tardiness-related scheduling problems whose principal property is not only the quality of the results, but also to give an ordering of the jobs (schedule) close to the optimal sequence. We list now some of the dispatching rules and heuristics that were selected to determine priorities, build schedules and contrast their outcomes with those obtained by the evolutionary algorithms.

WSPT (Weighted Shortest Processing Time first) the job with the highest importance per processing unit is selected first and in the final schedule jobs are ordered satisfying: $(w_1 / p_1) \geq (w_2 / p_2) \geq \dots \geq (w_n / p_n)$.
EDD (Earliest Due Date first) the job with earliest due date is selected first and in the final schedule jobs are ordered satisfying: $d_1 \leq d_2 \leq \dots \leq d_n$.
SLACK (Least slack) he job with smallest difference between due date and processing time is selected first first and in the final schedule jobs are ordered satisfying: $d_1\text{-}p_1 \leq d_2\text{-}p_2 \leq \dots \leq d_n\text{-}p_n$.

Hodgson Algorithm: This heuristic provides a schedule according to the following procedure,

Step 1: Order the activities in EDD order.
Step 2: If there are no tardy jobs left, 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: Revise the completion times and return to step 2.

The algorithm is optimal for the unweighted number of tardy jobs problem and can behave well for some instances of weighted number of tardy jobs.

Another high quality variant is known as the *Weighted Hodgson algorithm*, which modifies step 4 of the previous as follows:

Step 4: Move the single job j ($1 \leq j \leq k$) with the lowest w_i/p_i to the end of the sequence.

Rachamadagu and Morton Heuristic (R&M). This heuristic provides a schedule according to the following expression.

$$\pi_j = (w_j / p_j)[\exp\{-(S_j)^+ / kp_{av}\}]$$

with $Sj = [dj - (pj + Ch)]$ is the slack of job j at time Ch, where Ch is the total processing time of the jobs already scheduled, k is a parameter of the method (usually k =2.0) and p_{av} is the average processing time of jobs competing for top priority. In the *R&M* heuristic, also called the *Apparent Tardiness Cost* heuristic, jobs are scheduled one at a time and every time a machine becomes free a ranking index is computed for each remaining job. The job with the highest-ranking index is then selected to be processed next.

4. Studs and immigrants

A further MCMP extension is known as MCMP-SRI (stud and random immigrants) [9]. 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 and inserted in the mating pool. The number of parents in the mating pool is completed with randomly created individuals (random immigrants). The stud mates every other parent a number of times. The best offspring is inserted in the new population.

As EAs are blind search methods our new variant (MCMP-SRSI) [10], 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 intended to solve the scheduling problem 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, created by means of different heuristics. In this way the evolutionary algorithm incorporates problem-specific-knowledge supplied by the specific heuristic.

5. Current and future work

A number of experiments has been carried. For weighted tardiness benchmarks were retrieved from the OR-library, but for average tardiness and weighted number of tardy jobs we could not find available benchmarks. Consequently, we built our own test suite with data (p_j, d_j, w_j) extracted from 25 selected instances of the OR-library benchmarks for the weighted tardiness problem, with 40-jobs problem size, [1,2]. This data was the input for dispatching rules, conventional heuristics and our proposed multirecombined EAs, MCMP-SRI and MCMP-SRSI.

Results can be summarized as follows:

In the *weighted tardiness* problem:
Both evolutionary approaches compete with conventional heuristics, providing in most cases better solutions. When comparing MCMP-SRSI and MCMP-SRI, the former show smaller or equal mean percentile error in all instances, higher hit ratio, and smaller computational effort to find the best solution than the later.

.
In the *average tardiness* problem:

Both EAs produced solutions of higher quality (11% in average) than those achieved by typical heuristics. Again MCMP-SRSI outperforms MCMP-SRI. In particular its superiority is strongly related to speed of convergence (3 times faster in average

In the *weighted number of tardy jobs* problem:

MCMP-SRI behaves better in 50 % of the cases and worst in the rest when it is compared with the typical heuristics. Its average hit ratio is 73%. MCMP-SRSI shows improvements in 55% of the cases while reaching upper bounds in the rest of the cases. This behaviour is observed on every run and consequently the hit ratio is of a 100%. As indicated by the results MCMP-SRSI outperforms MCMP-SRI in quality of results and convergence rate. In particular its speed of convergence is 118 times faster in average than that of MCMP-SRI.

Further work will be dedicated to find better parameter settings and alternative ways to guide the evolutionary search for different scheduling problems.

6. References

[1] J.E.Beasley "Common Due Date Scheduling", 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, 341-350. 1998.

[3] T.Chen and M.Gupta, "Survey of scheduling research involving due date determination decision", *European Journal of Operational Research*, vol 38, pp. 156-166, 1989.

[4] A.E. Eiben, C.H.M. Van Kemenade, and J.N. Kok, "Orgy in the computer: Multi-parent reproduction in genetic algorithms". Proceedings of the *3rd European Conference on Artificial Life*, Springer-Verlag, 1995, number 929 in LNAI, pages 934-945.

[5] A.E. Eiben and. Th. Bäck, "An empirical investigation of multi-parent recombination operators in evolution strategies". Evolutionary Computation, 5(3):347-365, 1997.

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

[7] S. Esquivel, H. Leiva,.R. Gallard, "Multiple crossovers between multiple parents to improve search in evolutionary algorithms", Proceedings of the *Congress on Evolutionary Computation (IEEE)*. Washington DC, 1999, pp 1589-1594.

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

[9] D. Pandolfi, G. Vilanova, M. De San Pedro, A. Villagra, "Multirecombining studs and immigrants in evolutionary algorithm to face earliness-tardiness scheduling problems". Proceedings of the *International Conference in Soft Computing*. University of Paisley, Scotland, U.K., June2001, pp.138

[10] D. Pandolfi; G. Vilanova; M. De San Pedro; A. Villagra, R. Gallard: "Solving the Single-Machine Common Due Date Problem Via Stud and Immigrants in Evolutionary Computation", *World Multiconference on Systemics, Cybernetics and Informatics,* Orlando July 2001 pp 409-413

[11] M. Pinedo, "*Scheduling: Theory, Algorithms and System.*" First edition Prentice Hall, 1995.

[12] R.V. Rachamadugu, T.E. Morton, "Myopic heuristics for the single machine weighted tardiness problem". *GSIA*, Carnigie Mellon University, Pittsburgh, PA. 1982., Working paper 30-82-83.

[13] C. Reeves, "A genetic algorithm for flow shop sequencing", *Computers and Operations Research*, vol 22, pp5-13, 1995.

[14] Y. Tsujimura, M. Gen, E. Kubota: "Flow shop scheduling with fuzzy processing time using genetic algorithms". *The 11th Fuzzy Systems Symposium,* Okinawa,. 1995,.pp 248-252.