A Hybrid Genetic Algorithm with Local Search Approach for E/T Scheduling Problems on Identical Parallel Machines

Rainer Amorim
PPGI/IComp, UFAM
Manaus-AM, Brazil
rainer@icomp.ufam.edu.br

Bruno Dias
PPGI/IComp, UFAM
Manaus-AM, Brazil
bruno.dias@icomp.ufam.edu.br

Rosiane de Freitas
Institute of Computing, UFAM
Manaus-AM, Brazil
rosiane@icomp.ufam.edu.br

Eduardo Uchoa
Production Engineering, UFF
Niteroi-RJ, Brazil
uchoa@producao.uff.br

ABSTRACT

This work considers scheduling problems on single and parallel machines with arbitrary processing times and independent jobs, to minimize the sum of earliness-tardiness penalties. A Genetic Algorithm with a smart local search approach is presented, a 2-opt neighborhood-based with GPI moves and tie-breaking criteria, in a single sequence representation for single and multi-machine instances. Computational experiments are performed on Tanaka's instances for single machine, achieving all optimal solutions obtained by an IP exact method, for 40, 50, and 100 jobs. Moreover, our method is also suitable for dealing with multi-machine instances, achieving good solutions in a reasonable execution time, for 40, 50, and 100 jobs, with 2, 4, and 10 machines.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Global Optimization; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Sequencing and Scheduling; I.2.8 [**Artificial Intelligence**]: Heuristics

Keywords

due date penalties, heuristics, local search, scheduling

1. INTRODUCTION

Scheduling problems that involve due dates and deadlines are among the most difficult, studied, and applied in the real-world applications [4] [3]. Those involving earliness and tardiness penalties (E/T scheduling) have received particular attention since the adoption of *Just-in-Time* (JIT) concept, which relates to production without slack, neither before nor after the due date. In this work, we tackle with the minimization of both earliness and tardiness penalties simultaneously ($P||\sum \alpha_j E_j + \sum \beta_j T_j$), on m identical parallel machines P_i, $i = \{1, ..., m\}$, and n independent jobs J_j, $j = \{1, ..., n\}$ with arbitrary processing times p_j, a suggested time to finish - *due date* d_j, a completion time C_j, and arbitraries earliness and tardiness weights, α_j and β_j, respectively. Let E_j and T_j represented the *earliness* and *tardiness*, respectively, of *Job J*. A job J_j is early if $C_j \leq d_j$, and its earliness is given by $E_j = max\{0, d_j - C_j\}$. A J_j is tardy if $C_j > d_j$, and its tardiness is given by $T_j = max\{0, C_j - d_j\}$.

In this work we propose a hybrid Genetic Algorithm with a smart local search approach, involving a 2-opt neighborhood-based with GPI moves and tie-breaking criteria, in a single sequence representation for single and multi-machine instances, to solve the $P||\sum \alpha_j E_j + \sum \beta_j T_j$ scheduling problem, as in the following sections.

2. HYBRID GENETIC ALGORITHM

A hybrid Genetic Algorithm is proposed, which combines *Genetic Algorithm* (GA) and *Local Search* (LS) (**GA+LS** method). This algorithmic strategy uses a single sequence representation for multi-machine schedules, which simplifies the treatment of the problem, involving a 2-opt neighborhood-based with smart GPI moves and tie-breaking criteria.

The single sequence representation is treated as a chromosome for genetic operators of *mutation* and *position-based crossover*, proposed by Liu et al. [2], where two parents (chromosomes) are selected by *tournament selection* and the positions (genes) of that parents are randomly chosen to be fixed in the new chromosome (this operator preserves the order of the jobs on the machines, which is considered a good trait to evolve). For every child generated, a 2-opt based local search is performed in order to generate good solutions for the next population of the algorithm. A *tournament selection* is also used for *mutation* operator, but only one individual is selected, where the genes are randomly chosen for exchange. Our *local search* algorithm is based on the *Iterated Local Search* (ILS) method proposed by Rodrigues et al. [1]. There ensuring convergence monotonically increasing, since the best solutions generated in a population are taken directly to the next one, by applying the *elitism* operator.

3. COMPUTATIONAL EXPERIMENTS

The best configuration of the hybrid Genetic Algorithm proposed involved a population of nm individuals and nm iterations (n is the number of jobs and m is the number of machines). The iterations was divided in two parts, in the first part (1/3 of iterations) each new generated population was composed by 5% of elitism, 50% of mutation, 20% of

Table 1: Results for 40, 50 and 100 jobs, and 1, 2, 4 and 10 machines.

J.	Inst.	Optimal solutions				Hybrid Genetic Algorithm															
		1 m [5]	2 m	4 m	10 m	Best Solution				Average Solution				Iterations				Total Time			
						1 m	2 m	4 m	10 m	1 m	2 m	4 m	10 m	1 m	2 m	4 m	10 m	1 m	2 m	4 m	10 m
	1	54640	26063	11985	3988	**54640**	**26063**	**11985**	**3988**	54640	26063	11989	3988	1	2	3	1	1.9	9.3	57.2	990.3
	31	23291	11679	5950	3558	**23291**	**11679**	**5950**	**3558**	23291	11679	5950	3558	1	1	1	1	4.2	9.4	62.7	745.8
40	61	21737	12472	7714	5985	**21737**	**12472**	**7714**	**5985**	21737	12478	7714	5985	1	2	1	1	2.4	9.8	60.0	757.0
	91	48311	26309	15678	9818	**48311**	**26309**	**15678**	**9818**	48311	26309	15678	9818	1	1	1	1	2.7	9.9	57.3	742.9
	121	122538	64584	35783	18818	**122538**	**64584**	**35783**	**18818**	122538	64584	35783	18818	1	1	1	1	3.3	10.2	75.3	743.2
	1	130548	62985	29222	9006	**130548**	**62985**	29229	9154	130548	62986	29235	9154	1	4	9	4	8.5	30.9	183.6	2390.6
	31	37565	18266	8709	3839	**37565**	**18266**	**8709**	**3839**	37620	18266	8710	3839	1	1	5	2	9.7	35.9	158.9	2336.3
50	61	34640	17456	9376	5856	**34640**	**17456**	**9376**	**5856**	34654	17504	9376	5856	1	2	2	1	5.4	30.1	160.4	2250.7
	91	89715	47513	26659	14527	**89715**	**47513**	**26659**	**14527**	89732	47513	26660	14527	1	2	2	1	5.9	27.5	155.9	3089.4
	121	79007	41989	23884	13346	**79007**	**41989**	**23884**	**13346**	79007	41989	23885	13346	1	1	3	1	8.1	28.7	152.7	2307.5
	1	405122	198281	94951	31489	**405122**	**198281**	94963	33273	405122	198284	94964	33273	3	7	15	23	158.6	907.2	4840.5	78822.6
	31	193480	95038	–	17012	**193480**	**95038**	45750	17030	193480	95059	45752	17030	2	5	11	15	168.9	1077.5	5050.2	85821.2
100	61	102185	52739	–	14634	**102185**	52740	28273	14643	102185	52740	28280	14634	2	4	8	15	179.5	906.0	5379.0	88134.1
	91	249743	130165	70755	35784	**249743**	**130165**	**70755**	35791	249746	130166	70755	35791	2	3	5	7	187.4	842.6	4844.4	89648.2
	121	471761	242234	127686	59347	**471761**	**242234**	127702	59350	471771	242235	127710	59350	2	6	19	15	174.1	825.7	4769.7	69938.6

position-based crossover and 25% of position-based crossover followed by Local Search, where the goal was to better explore the feasible region of our E/T scheduling problem getting the widest range of possible solutions, and increasing the diversity of individuals. In the second part (2/3 of iterations), each new generated population was composed by 10% of elitism, 30% of mutation, 10% of position-based crossover, 50% of position-based crossover with Local Search, where the goal was to better explore good regions of solutions, over the populations generated by the first part, the feasible region find optimal or near-optimal solutions, in an attempt to find the global optimal solution, and guaranteeing at least the generation of multiple local optima solutions, and multiple high quality solutions in general to the problem. When the algorithm reaches the number of iterations, nm iterations is added to the current number of iterations, searching for better solutions. The process continues, being held nm new iterations every time at least one new solution is generated. When no better solution is found, then the evolutionary process has stalled, but having converged to good solutions to the problem.

The GA+LS was tested with Tanaka [5] [1] instances. We adapted these instances in order to test them on identical parallel machines, by dividing the due dates by the number of machines. The results of the algorithm proposed for single and parallel machines is presented in the Table 1, where is presented the results for 40, 50 and 100 jobs on 1, 2, 4 and 10 machines. For single machine, the results were compared with Tanaka [5] optimal solutions, achieving all of them. For multi-machines, we have adapted the OR-Library instances available for the weighted tardiness scheduling problem (such as done by Tanaka for single machine) [2], dividing the processing time of a job by the amount of available machines, involving 2, 4, and 10 machines.

The optimal solutions presented for multi-machines was obtained by using *Branch-and-Cut* algorithm by IBM/ILOG CPLEX 12.4 solver, where our proposed formulation was implemented by using the UFFLP library [3] with C++ where, given and instance for the problem, it is possible to generate the mathematical model to be executed in CPLEX. The GA+LS method achieved optimal solutions in most cases tested. The following columns presents the best solution obtained (*Best Solution*), the average of three executions for every instance (*Average Solution*), quantity of iterations (*Iterations*) and the *Total Time* of the algorithm.

4. CONCLUDING REMARKS

This work considered the earliness-tardiness scheduling problem on single and identical parallel machines, and independent jobs with arbitrary processing times, that is, $P || \sum \alpha_j E_j + \sum \beta_j T_j$ in the 3-field notation. We proposed a hybrid algorithmic strategy involving Genetic Algorithm with a smart Local Search approach (a 2-opt neighborhood-based with GPI moves and tie-breaking criteria), in a single sequence representation for single and multi-machine instances. The proposed algorithm was able to achieve optimal solutions for single machine Tanaka [5] instances. The proposed method was also tested with multi-machine instances, achieving good solutions in a reasonable computational time for 40, 50, and 100 jobs, and 2, 4, and 10 machines. Ongoing works involve the developing of a hybrid exact method based on a proposed integer programming formulation.

5. ACKNOWLEDGMENTS

This work is partially supported by brazilian agencies, which are CAPES, FAPEAM, FAPERJ, and MCT/CNPq Universal research projects of number 473513/2011-0 and number 479275/2011-4.

6. REFERENCES

[1] R. de FreitasRodrigues, A. Pessoa, E. Uchoa, and M. P. de Aragão. Heuristic algorithm for the parallel machine total weighted tardiness scheduling problem. *Relatórios de pesquisa em engenharia de produção*, 8:1–12, 2008.

[2] N. Liu, M. Abdelrahman, and S. Ramaswamy. A Genetic Algorithm for Single Machine Total Weighted Tardiness Scheduling Problem. *International Journal of Intelligent Control and Systems*, 10:218–225, Sep. 2005.

[3] A. Pessoa, E. Uchoa, M. P. de Aragão, and R. de FreitasRodrigues. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Mathematical Programming Computation*, 2:259–290, 2010.

[4] M. L. Pinedo. Scheduling: Theory, algorithms, and systems. *Springer Publishing Company, Incorporated*, 4a ed.:1–104, 2012.

[5] S. Tanaka. An exact algorithm for the single-machine earliness-tardiness scheduling problem. *Springer Optimization and Its Applications*, 60:21–40, 2012.

[1]http://turbine.kuee.kyoto-u.ac.jp/∼tanaka/SiPS/index.html
[2]http://people.brunel.ac.uk/ mastjjb/jeb/info.html
[3]http://www.gapso.com.br/ufflp