remote
9th LAWCG
and
MDA
November 25th, 2020

A Matheuristic Approach for the Weighted Just-in-Time Scheduling Problem

AMORIM, Rainer X. de – ICET/UFAM – raineramorim@ufam.edu.br, DE FREITAS, Rosiane – ICOMP/UFAM – rosiane@icomp.ufam.edu.br, PESSOA, Artur - artur@producao.uff.br, UCHOA, Eduardo - uchoa@producao.uff.br

INTRODUCTION

This work presents a hybrid exact-heuristic algorithmic approach, based on an arc-time indexed mixed-integer programming formulation and a generalized evolutionary based on a strong local search, in order to better solve the problem $P||\sum\alpha_j E\_j + \beta_j T_j$ (WET). The selected arcs from local optimal solutions generated by a Genetic Algorithm based on a strong Local Search (GLS), are given as input to an IP Arc-time indexed formulation, which is solved to produce better solutions at CPLEX. The proposed Hybrid Matheuristic method is capable to produce better results when compared with the previous best results in the literature.

OBJECTIVE

The objective of this work is to develop an exact-heuristic method to solve large instances of the the identical parallel machine Weighted Just-in-Time Scheduling Problem.

JUST-IN-TIME SCHEDULING PROBLEM

Considering the classical NP-hard parallel-machine weighted earliness-tardiness scheduling problem, $\boldsymbol{P||\sum\alpha_j E_j + \sum\beta_j T_j}$ **(WET)**, in 3-field notation [1], where $\boldsymbol{j = \{1, \ldots, n\}}$ is the set of independent jobs to be processed without preemption, in $\boldsymbol{m}$ **identical parallel machines**, where each one can process at least one job on a given time. Every job j has a positive processing time p_j, a due date d_j and a positive earliness (α_j) and tardiness (β_j) weights. The earliness of a job is defined as $E_j = \max\{0;\ d_j - C_j\}$ and the tardiness of a job is defined as $T_j = \max\{0;\ C_j - d_j\}$, where C_j is the completion time of the job [2]. Figure 1 (a) presents an example of 8 jobs for the problem followed by a solution representation for single machine scheduling in Figure 1 (b) and its corresponding representation for identical parallel machines in Figure 1 (c), considering three identical parallel machines.

J	p_j	d_j	α_j	β_j	C_j	E_j	T_j	$\alpha_j E_j$	$\beta_j T_j$
j_1	3	3	3	5	3	0	0	0	0
j_2	2	6	4	5	2	4	0	16	0
j_3	3	6	8	8	3	3	0	24	0
j_4	4	4	8	10	6	0	2	0	12
j_5	6	6	6	4	9	0	3	0	27
j_6	7	10	7	3	10	0	0	0	0
j_7	3	11	4	2	10	1	0	4	0
j_8	3	8	5	8	12	0	4	0	48

$\sum\alpha_j E_j = 44$ and $\sum\beta_j T_j = 87$

(a)

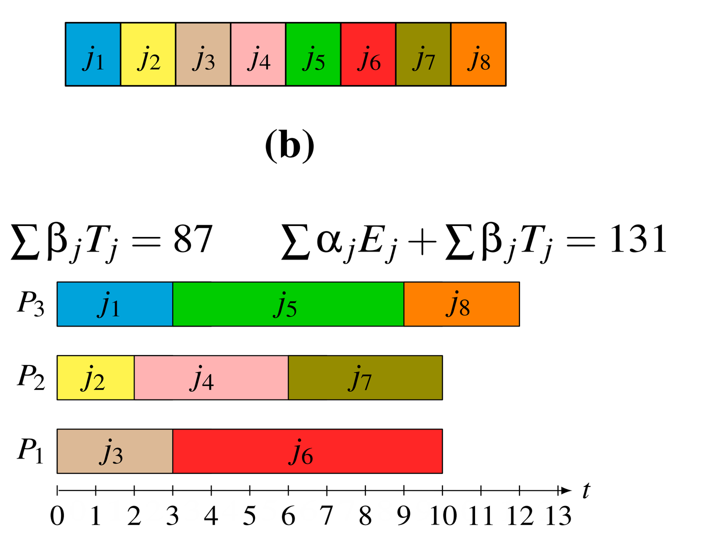

Figure 1: (a) An instance example of 8 jobs for the weighted tardiness and earliness-tardiness scheduling problem. Scheduling examples for the (b) identical parallel machines using machine-oriented Gantt chart and (c) the single sequence representation.

THE HYBRID MATHEURISTIC

The Hybrid Matheuristic (MathGLS-IP) is based on two steps:

- STEP 1: Heuristic approach (GLS)

The best local optimal solution generated by the GLS (Figure 2) is kept in a Hash Table on every generation, which will be used as a selected set of arcs to the IP Arc-time formulation. A solution representation of the Arc-time is presented in Figure 3.

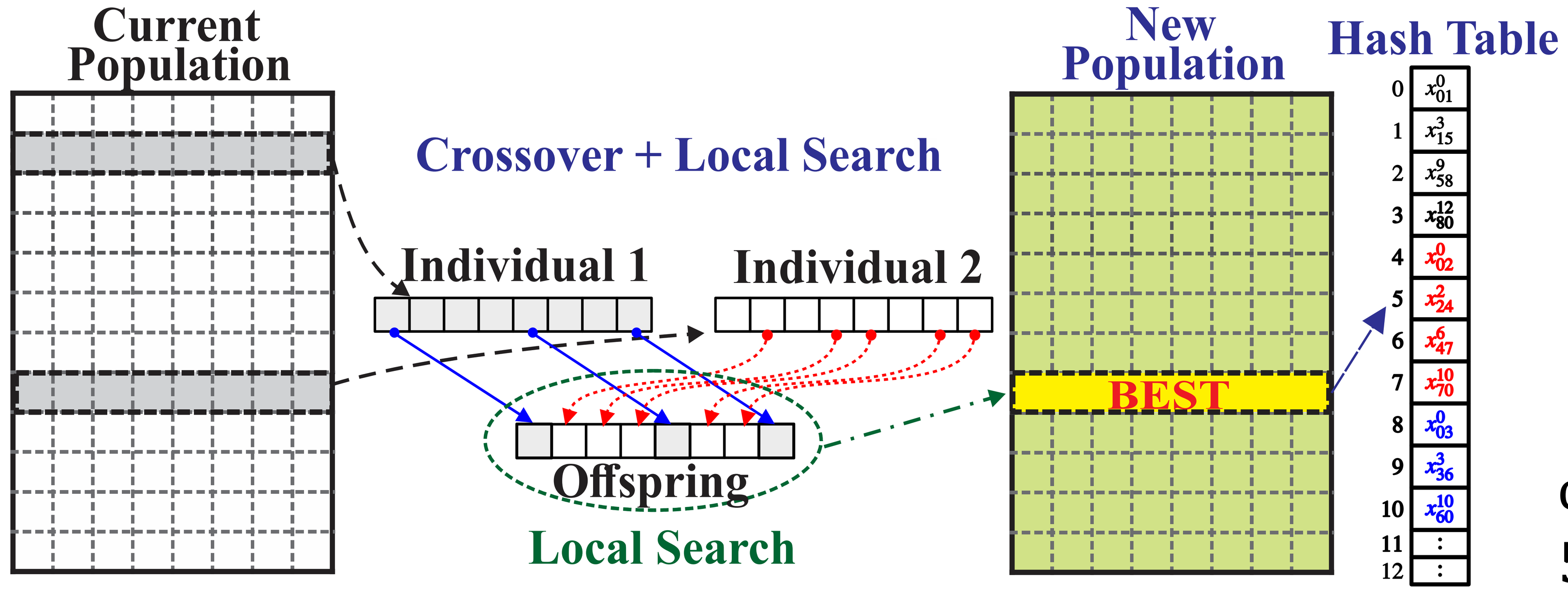

Figure 2: GLS - Crossover operation followed by local search, where the best local optimal solution of every generation is kept in a Hash Table.

- STEP 2: Exact approach (Solving the Arc-time)

When GLS procedure finishes, the selected arcs kept in the Hash Table are used to build the Arc-time, and then, solve it in CPLEX to get better convergence or improve the solution for a given instance of the problem. The Arc-time indexed formulation, proposed by Pessoa et al. [3], is presented bellow. The MathGLS-IP method eliminates the Constraints (4), in order to decrease the number of binary variables of idle time at the end of a scheduling.

$\sum\alpha_j E\_j + \sum\beta_j T_j$

Every job must be visited by exactly one path

$$\text{Minimize} \quad \sum_{j\in J_+}\sum_{j\in J\setminus\{i\}}\sum_{t=p_i}^{T-p_j} f_j(t+p_j)\,x_{ij}^t \qquad (1)$$

$$\text{s.t.} \quad \sum_{j\in J_+}\sum_{t=p_i}^{T-p_j} x_{ij}^t = 1 \quad (\forall j \in J) \qquad (2)$$

Define the network flow of m units over an acyclic layered graph $G = (V,A)$

$$\sum_{\substack{j\in J_+\setminus\{i\},\\ t-p_j\ge 0}} x_{ji}^t - \sum_{\substack{j\in J_+\setminus\{i\},\\ t+p_i+p_j\le T}} x_{ij}^{t+p_i} = 0 \quad (\forall i\in J;\ t = 0,\ldots,T-p_i) \qquad (3)$$

Eliminated Constraints

$$\sum_{\substack{j\in J_+,\\ t-p_j\ge 0}} x_{j0}^t - \sum_{\substack{j\in J_+,\\ t+p_j+1\le T}} x_{0j}^{t+1} = 0 \quad (t = 0,\ldots,T-1) \qquad (4)$$

$$\sum_{j\in J_+} x_{0j}^0 = m \qquad (5)$$

Binary variables

$$x_{ij}^t \in Z_+ \quad (\forall i\in J_+;\ \forall j\in J_+\{i\};\ t = p_i,\ldots,T-p_j) \qquad (6)$$

$$x_{00}^t \in Z_+ \quad (t = 0,\ldots,T-1) \qquad (7)$$

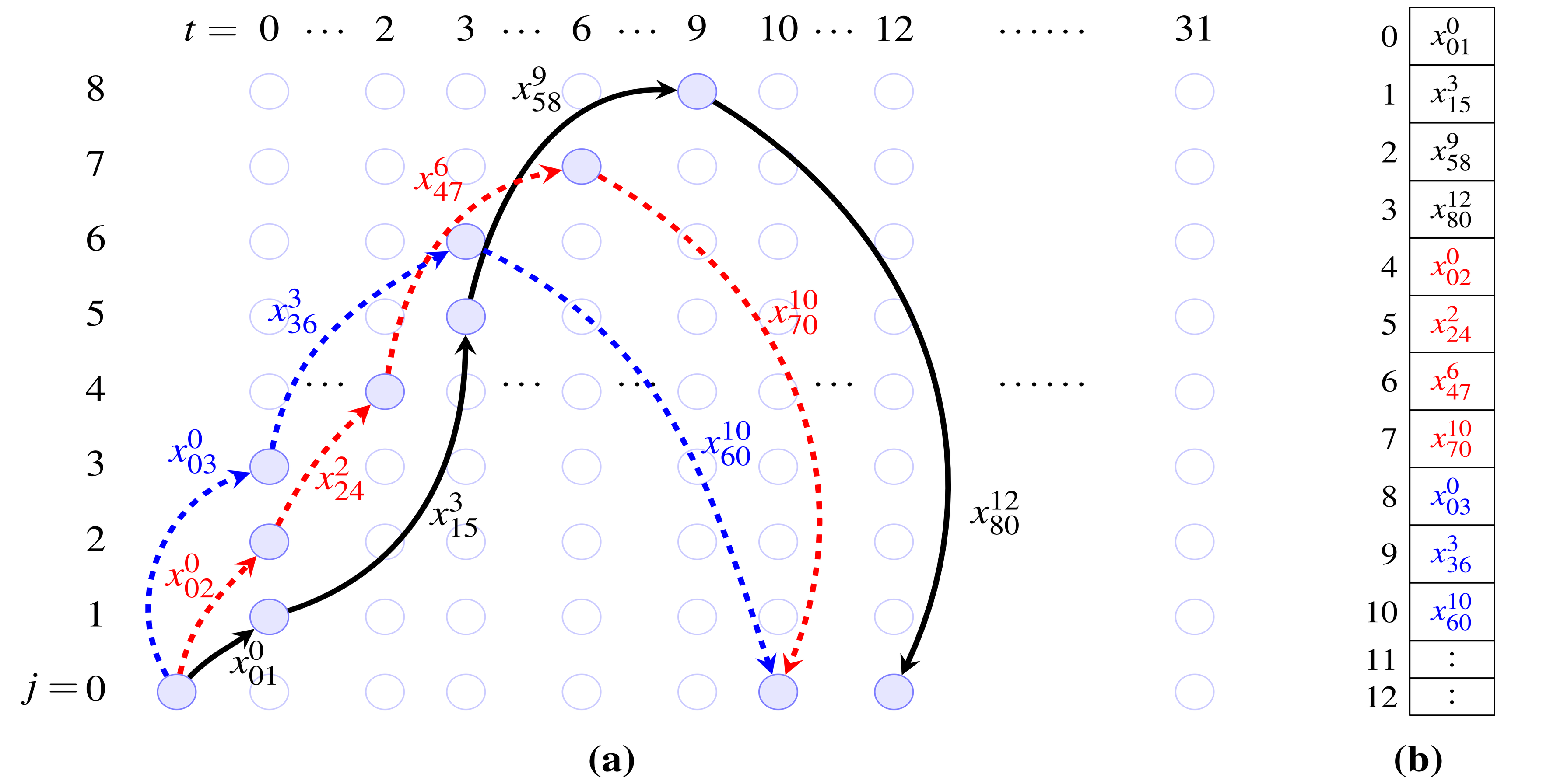

Figure 3: (a) Parallel machine network flow representation for the solution in Figure 1 (c) and the stored arcs from this solution in a hash table presented in (b) (we keep a set of stored solutions - not only one solution).

COMPUTATIONAL EXPERIMENTS

In Table 1 we present a resume of the computational experiments, compared with the literature. MathGLS-IP solves large instances up to 500 jobs and 2, 4 and 10 identical parallel machines. Our method also presents results for 200 instances, not yet known in the literature, and improved 4. Detailed results can be observed at Amorim [4].

Table 1: Computational Experiments compared with the Literature for the problem $P||\sum\alpha_j E_j + \sum\beta_j T_j$ with 40, 50 and 100 jobs on 2-10 machines.

Instance group	Kramer and Subramanian [5]				MathGLS-IP				
	Best run		Average		Best run			Average	
	GAP (%)	BKS	GAP (%)	Time (s)	GAP (%)	#	*	GAP (%)	Time (s)
wet40-2m	0,000	12	0,000	5,592	0,000	24	0	0,001	6,420
wet40-4m	0,000	12	0,001	6,258	0,000	24	0	0,001	9,261
wet40-10m	0,000	5	0,000	4,080	0,000	25	0	0,002	8,882
wet50-2m	0,000	11	0,001	12,617	0,000	23	0	0,000	13,623
wet50-4m	0,000	12	0,306	14,145	0,000	25	0	0,004	15,939
wet50-10m	0,000	5	0,014	9,320	0,000	24	0	0,038	13,686
wet100-2m	0,000	12	0,008	168,483	0,000	24	0	0,004	166,087
wet100-4m	0,790	6	0,858	190,309	0,000	23	4	0,055	114,163
wet100-10m	0,161	0	0,227	140,380	0,089	8	0	0,332	102,153
Total	---	**75**	---	---	---	**200**	**4**	---	---
Average	0,106	---	0,157	61,243	0,010	---	---	0,049	50,024

BKS – Amount of Best Known Solutions in the literature   * – Amount of improved solutions
– Amount of solutions equal to BKS

REFERENCES

[1] GRAHAM, R. L.; LAWLER, E. L.; LENSTRA, J. K.; and RINNOOY KAN, A. H. G. Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics, 5:287-326, 1979.
[2] PINEDO, M. L. Scheduling: Theory, algorithms, and systems. Springer Publishing Company, Incorporated, 4a ed.:1-104, 2012.
[3] PESSOA, A.; UCHOA, E.; ARAGÃO, M. P. de; and DE FREITAS, R. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. Mathematical Programming Computation, 2(3-4):259-290, 2010.
[4] AMORIM, R. Estratégias Algorítmicas Exatas e Híbridas para Problemas de Escalonamento em Máquinas Paralelas com Penalidades de Antecipação e Atraso. Tese (Doutorado em Informática), 2017.
[5] KRAMER, A.; SUBRAMANIAN, A. A unified heuristic and an annotated bibliography for a large class of earliness–tardiness scheduling problems. Journal of Scheduling, 22(1): 21-57, 2019.

ACKNOWLEDGMENT

UFAM ICOMP Instituto de Computação ICET uff Universidade Federal Fluminense CAPES FAPEAM