AN HEURISTIC ALTERNATIVE TO OPTIMIZE MAKESPAN OVER M PARALLEL MACHINES WITH RELEASE AND SECUENCE-DEPENDENT SETUP TIMES

$P_m \mid r_j, s_{jk} \mid C_{max}$

LOPEZ, Julián , ROZO, Viviana , SAAVEDRA, Iván and
PATERNINA-ARBOLEDA, Carlos D.
Universidad del Norte, Department of Industrial Engineering
Barranquilla, Colombia

ABSTRACT: Scheduling parallels machines problems with release and sequence-dependent setup times are considered NP-Hard problems because of their high algorithmic complexity. The propose of this article is to show a heuristic alternative the minimization of makespan satisfying all last restrictions. To make this possible, is propose an heuristic that make Jobs sequence and allocation in all machines at the same time looking for a scheduling that minimize release times, taking care the disponibility of jobs times to get them started for be scheduled and minimizing the makespan. Analyzing the effects of the heuristic, there were some changes in the propose heuristic about the selection criterion of the first job to be schedule en each machine whit the expectative of make comparisons taking like metric of evaluation the makespan for each instances of test.

KEY WORDS: Heuristic, parallels machines, makespan, release times, setup times.

INTRODUCTION

In the last years scheduling problems had make some emphasis on parallels machines problems, in the same way their is and diversity of optimization problems, where a lot of them are classified like NP-Hard. For this kind of problems, some alternatives of combinatory exploration are considered that allow generate optimal solutions given a feasible solution space.

From this point of view, the goal is to obtain heuristic alternatives of construction that generate feasible solution spaces which they could be optimize with meta-heuristic process or with randomly combinations using improvement heuristics.

In this order of ideas, the propose of this article is to present an heuristic alternative that show a scheduling that minimize the makespan for a dynamic equals parallels machines with release and sequence-dependent setup times. The critic point over the alternative propose is the criteria of selection of the first jobs to be Schedule in each machine, for this reason was realized a variation on this criteria with the objective of compare the results given by the initial criteria.

DEFINITION AND PROBLEM FORMULATION

The problem treat don this study is the scheduling of N Jobs, with N ∈ [15, 20] on a station with m equals parallels machines with m ∈ [3,5]. The Jobs have release and sequence-dependent setup times, processing times, and arrive times. The Jobs can be process on whichever machine and each machine has to finish completely whit the process of a Job before start to process the next Job. The problem is given by:

$\mathbf{P_m \mid r_j, s_{jk} \mid C_{max}}$; m ∈ [3,5]; j, k = {1, 2, ..., N}, N ∈ [15, 20]. The problem treated can be formulated using the next and given a definition of some variables and restrictions:

Pm: Equals parallels machines

Cmax: Makespan

R_k: Release times

S_{jk}: Setup times from the Job j to the Job k.

Objetive function:

Minimize Cmax.

$\overline{P_m}$: Array to store the summatory of all release time and setup times of the Jobs scheduled on machine m.

$\overline{J_{am}}$: Array to store the sequence of Jobs scheduled on machine m.

$\overline{J_{cm}}$: Array to store all possible Jobs to be schedule by machine m, they have to satisfy the restriction (1).

$\overline{J}$: Array to store all Jobs that aren't schedule yet.

Following these restrictions:

$$\overline{P_m} + S_{jk} \geq r_k, \forall m = 1 \ldots M, \forall k \in \overline{J} \quad (1)$$

With the next criteria is selected the Job to be schedule on machine m and is allocated on $\overline{J_{am}}$:

$$Min\ (\overline{P_m} + S_{jk} + P_k), \forall K \in \overline{J_{cm}}$$

And $\overline{P_m}$:

$\overline{P_m} = \overline{P_m} + S_{jk} + P_k$, being k the next Job to be Schedule on machine m.

HEURISTIC FORMULATION JIV

The algorithm design to give a solution for the problem under study is base on the allocation and sequence of the Jobs over the different machines. The objective is to propose an algorithmic alternative different to the existents on the scientist literature, Taken care that for this kind studies we can find heuristics based first on sequence and next on allocation or vice versa.

The heuristic design, use the next parameters of initialization and operation:

The array of all Jobs to be Schedule is initialized first, $\overline{J} = \{1,2,\ldots,N\}$, being N the number of Jobs to be schedule. The arrays $J_a, J_{cm}, P_m = \Phi$. S_{ij} Is started and this array store the setup times for going of the Job i to Job j on a machine.

The sort of the release times is done, in the way of the array J have the next characteristic: $r_1 \leq r_2 \leq \cdots \leq r_n$

1.) The m first Jobs are obtained in J and candidates are generated by machine with the objective of Schedule some ones that have less release time. On this way, $\sum$ of the makespan by machine is carry $\overline{P_m} = r_m + p_m, m \in \{1,\cdots,M\}$, to compare between deferent's iterations; after that the job is schedule on machine m ($J_{am} = j_m$) and is eliminated from J.
2.) Until exist Jobs that are not schedule on J, for all machines is make a search for those candidates that generate the less makespan if they are Scheduled by each iteration on machine m. A candidate Job must satisfy :

$$\overline{P_m} + S_{jk} \geq r_k, \forall m = \{1\ldots M\}, \forall k \in J \qquad (1)$$

If there are no Job for this machine that satisfy this condition, the Job with less r_k is selected on J and is allocated like a candidate for this machine.
3.) The Job and machine that generate the less makespan by iteration is selected, alter this is schedule:

$J_a = j_{km}$, where $j_{km} \in J_{cm}$, j_{km} make the makespan minimum on machine *m* for the iteration.
4.) The Job is deleted from J and is carried for the machine accumulated, the release and setup time, $\overline{P_m} = \overline{P_m} + S_{jk} + P_k$
5.) In the end, if $J = \Phi$ the grater C between all machines is selected and is accumulated on the $C_{\max}$.

```
FUNCTION JIV_RELAXED
BEGINING
M ,N// machines and Jobs
Inicialización( N );// Parameters
SortRi( J );// Sort jobs by release time
Select_first( J );//select first m jobs for  Schedule
While( J != NULL )
   /*Candidates to be schedule are obtained for all
      machines */
   If( \overline{P_m} + S_jk >= r_k )
     Min (\overline{P_m} + S_jk + P_k), k ∈ \overline{J_cm}
   /*SINO
     Min (R_k + P_k), k ∈ \overline{J}
   /*The candidates are evaluated and the Job that
      make the makespan minimum on machine m */
   Ja_m=Min_m; J-=Ja_m; //The Job is allocated and
      deleted from J
   Pm=Pm+Sik+Pk;
End While
Select_Cmax( Pm, Ja_m) /*The C max is selected
END
END-FUNCTION
```

INSTANCES OF TEST AND EXPERIMENTAL RESULTS

The heuristic propose with each of its variations on the selection criteria of the first machines schedules was proved for 28 instances different configured like is presented:

$r_j \geq 0$

$P_j > 0$

$S_{ij} \geq 0$

N jobs, with N ∈ [15, 20]

M equals parallels machines with M ∈ [3,5].

The variation used over the selection criteria for the alternative presented consisted on organize the Jobs that satisfy with *m* less r_j using LPT and next allocating the Jobs to the *m* machines in order obtained by LPT.

On next is presented the experimental results of the heuristic on study. The instances used are organized in the order propose on the magnetic support of this article. (Attached 1)

Table1. Results of Cmax by instances of evaluation

INST	HEURISTIC		INST	HEURISTIC	
	JIV RELAX	JIV VARIADO		JIV RELAX	JIV VARIADO
1	99	111	**15**	32	37
2	79	80	**16**	47	47
3	33	35	**17**	39	41
4	65	62	**18**	46	46
5	82	73	**19**	30	24
6	95	104	**20**	42	45
7	84	99	**21**	46	46
8	78	86	**22**	42	36
9	69	71	**23**	47	52
10	49	49	**24**	41	39
11	70	71	**25**	62	76
12	50	55	**26**	66	66
13	90	90	**27**	87	87
14	107	107	**28**	82	82

Figure1. Results of Cmax by instances of evaluation JIV RELAXED – JIV VARIED.

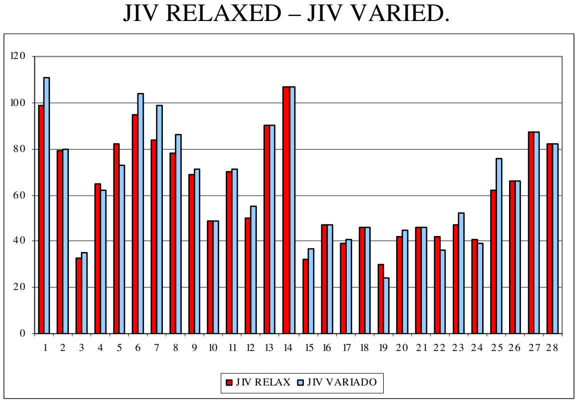

Table2. Desv. Value over instance test

INST	DESV. STANDAR	INST	DESV. STANDAR
1	8,49	**15**	3,54
2	0,71	**16**	0,00
3	1,41	**17**	1,41
4	2,12	**18**	0,00
5	6,36	**19**	4,24
6	6,36	**20**	2,12
7	10,61	**21**	0,00
8	5,66	**22**	4,24
9	1,41	**23**	3,54
10	0,00	**24**	1,41
11	0,71	**25**	9,90
12	3,54	**26**	0,00
13	0,00	**27**	0,00
14	0,00	**28**	0,00

CONCLUTIONS

How is show on the experimental results, there are assignations which the variations of the heuristic propose present the best makespan that the one study to this moment, but in proportion of makespan and differences between results, is observe that the initial alternative propose generate best stages for this problem its variation.

For instances too small that don't have more than 5 Jobs and 2 machines, the heuristic variant applied give best results compares with the initial propose. But, once the instance getting greater on complexity the initial propose keep stronger.

With no doubt , the conclusion is that is possible find very good results, maybe optimal, with the alternative propose, if a initial selection criteria is design that help to generate a optimal Cmax. For this kind of initial generation, it could be a interesting futures studies, generate this studies using heuristics of construction and randomly improvements (Meta-heuristics).

REFERENCIAS

[1] Mejia, G, Eliécer Gutierrez. Evaluation of genetic algorithms for solving the parallel machine scheduling problem with sequence dependent set up times and due

dates restrictions. III congreso colombiano y I conferencia andina internacional de investigación de operaciones.

[2] Salazar A. Maria A, Roger Z Rios. Minimización heurística del número de tareas tardías al secuenciar líneas de flujo. División de postgrado en Ingeniería de sistemas, FIME – UANL.

[3] Lopez De Haro, Santiago, P. Sanches Martín y Javier Conde. Secuenciación de tareas mediante meta-heurísticos. VIII Congreso de ingeniería de organización leganés.

[4] Baran, Benjamín, A. Hermosilla. Comparación de un sistema de colonia de hormigas una estrategia evolutiva para el problema del ruteo de vehículos con ventanas de tiempo en un contexto multi-objetivo.