MINIMIZATION OF Cmax IN PARALLEL MACHINES WITH IDENTICALL PARALLEL MACHINES, REALEASE AND SETUP TIMES

Celia Aleán Zapata, Vanesa Manotas Niño, Maria Orozco Arredondo,
Iván Ojito Castro, Carlos D. Paternina-Arboleda

Industrial Engineering Department
Universidad del Norte
Barranquilla, Colombia.
{calean, vmanotas, orozcoc, iojito, cpaterni}@uninorte.edu.co

Abstract:

In this paper we propose and show the applicability of a heuristic algorithm to determine the optimization of the makespan when 3 to 5 parallel machines make 15 to 20 works. The results showed by the algorithm are obtained from the interaction between the process time, the release date and setup times, being these the most common restrictions of all the scheduling problems. The main contribution of this paper is the development of a model that allows to identify the logical sequence of jobs to carry out for each machine in a way to obtain optimal results.

Key words: Optimization, makespan, parallel machines, process time, release date, setup time.

INTRODUCTION

In the practice, several basic objectives are had in consideration for the industries for to get optimal results as far as the scheduling of the production. One of these objectives is the minimizing of makespan, denoted by C_{max} and defined as the time the last job leaves the system.

In the case the parallel machines in which the machines are equal (conditions of speed, setups, ect) and each one is capable of processing any job and jobs have to be processed without interruption once processing starts, the minimizing of makespan looking for the way of properly balance the production over the machines and minimize the sum of the sequence dependent setup times to maximize the throughput rate. Is for these reason that is so important develop mechanisms that allow programmer to make agile the process of scheduling, of such form that this is effective before the situation presented in this paper that it works with 3 to 5 machines and 15 to 20 works made in machines in parallel.

PROBLEM DEFINITION

The problem treated in this paper is the programming of N works in a station with m identical machines in parallel. The jobs have times dependent of enlistment of the sequence and the release times, and each job has specifics process times. The jobs can be processed in anyone of the machines of the station. All the works are available in time zero and each machine must finish completely the processing of a work to initiate the processing of the following work, it means a non preemptive job.

The problem can be formulated using the following annotation and defining the following variables and restrictions:

P_j the process time of job j (j = 1,…, n).

r_j the release time of job j (j = 1,…, n).

S_{jk} the time of enlistment for work j if it is preceded by work k (j, k = 1,..., n).

DEVELOPMENT OF THE HEURISTIC

To minimize the criteria of makespan in problems of scheduling sequence dependent jobs on identical parallel machines, the heuristic was developed having in consideration the process time of each job j, P_j, the release dates (the times the job arrives at the system) of each job j, r_j, and the setup times (the machine preparation time depending of the sequence), S_{jk}, resolving the problem:

$$\mathbf{P_m \mid r_j, s_{j\,k} \mid C_{max}}$$

The tests of the algorithm took control of 3 to 5 machines making in parallel 15 to 20 works, where m is the number of parallel machines and N is the number of jobs:

$$\mathbf{m \in [3,5];\ j, k = \{1, 2, ..., N\},\ \ N \in [15, 20].}$$

The heuristic algorithm was developed in visual basic application for excel, contains a programmer interphase who asks for the data like show in the next example:

# Machines	# Jobs
3	20

Trabajo	Pj	Rj
1	6	0
2	3	0
3	4	0
4	7	4
5	4	8
6	6	10
7	6	15
8	4	19
9	5	25
10	4	30
11	6	0
12	4	0
13	5	0
14	5	4
15	4	8
16	5	10
17	7	15
18	4	19
19	5	25
20	4	30

Sjk	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		3	0	6	6	6	1	8	2	5	0	7	5	12	7	10	5	4	1	8
2	3		9	5	4	2	3	5	4	6	10	9	6	5	5	9	7	2	3	3
3	1	6		7	5	7	1	7	9	2	6	4	5	8	3	8	3	10	5	0
4	5	5	2		5	6	8	10	2	1	8	9	2	7	4	1	3	4	2	7
5	8	5	2	5		8	6	4	1	0	4	7	3	6	6	2	6	0	2	1
6	5	2	6	9	2		9	7	3	2	9	1	5	7	9	2	2	7	4	5
7	8	3	4	7	3	1		5	7	6	0	3	5	1	3	6	7	2	9	8
8	2	4	8	2	1	5	8		5	4	2	6	8	3	3	1	9	3	6	9
9	8	5	1	3	5	8	6	6		2	6	9	8	6	6	5	7	6	4	4
10	2	2	7	6	1	7	3	5	5		1	8	1	1	1	1	0	1	2	0
11	6	7	1	5	6	2	3	9	4	7		3	5	1	5	9	0	7	2	2
12	8	7	5	4	7	0	8	5	3	3	2		3	0	11	6	7	8	6	8
13	6	3	8	6	8	2	6	4	2	6	4	7		6	4	2	9	2	4	2
14	2	2	4	9	7	3	9	6	12	2	8	8	3		3	3	7	1	5	5
15	9	0	4	4	7	1	4	7	2	5	4	2	9	2		8	8	3	8	7
16	6	4	7	4	5	5	7	5	5	1	9	7	6	4	5		4	2	8	3
17	9	8	8	7	5	1	8	3	7	8	3	5	5	1	8	7		4	1	8
18	6	2	2	9	6	6	9	6	3	12	5	0	7	2	3	6	1		6	9
19	7	8	6	4	5	7	4	5	8	7	6	2	4	9	6	4	2	0		6
20	3	6	7	2	1	8	2	1	9	3	8	1	5	8	9	4	0	6	7	

Once the programmer indexes all the data, comes itself to execute the algorithm.

ALGORITHM

Algorithm Minimizing $C_{\max}$ $\quad P_m \left| r_j, s_{jk} \right| C_{\max}$

Step 1

Sort considering r_j in ascending order and broken ties P_J longest processing time (LPT)

Step 2

Prioritize the job: Select the job with the smallest r_j.

Step 3

Schedule the highest priority job on the machine capable of finishing it first.

In the case of a tie between two or more machines it is selected the one with lower S_{jk} to be assigned.

To select the machine and assign the next work it is computed the $C_{\max}$ for each one considering the following restrictions:

If $r_j > C_{\max\, before}$

$$C_{\max\, new} = P_J + r_j + s_{jk}$$

Else

$$C_{\max\, new} = P_J + s_{jk}$$

End

And looking for the minimum value

$$M_j \middle| C_j = \min(C_i) \forall i$$

Once executed the algorithm by means of the interphase of excel, for the previous example the following results were obtained:

M1	M2	M3
1	11	13
2	12	3
4	14	15
16	5	6
7	17	18
9	8	19
	20	10
67	**62**	**64**

RESULTS

This algorithm was proved with 28 instances generated in random form. The results obtained were the following:

	Cmax				
Instance	**M1**	**M2**	**M3**	**M4**	**M5**
1	90	60	76		
2	73	82	61	82	
3	36	36	37	39	
4	48	43	62	44	
5	79	92	82	93	85
6	90	72	57	57	
7	51	45	47	44	
8	52	54	57		
9	30	24	27	31	28
10	67	62	64		
11	48	56	46		
12	41	40	44	34	
13	116	91	88		
14	76	86	89	83	
15	86	74	81		
16	53	45	44	40	46
17	77	65	65	59	
18	54	46	46	51	59
19	91	59	57	61	53
20	99	100	97		
21	48	42	52	42	
22	43	33	36	38	32
23	143	153	156		
24	116	76	78		
25	35	35	36	35	39
26	79	80	68		
27	64	67	65	80	
28	92	74	73	76	78

CONCLUSIONS

The problem of scheduling tasks in machines widely has been studied through the years in order to optimize times of process, to diminish lates, among other reasons. Many kinds of problems with different variations exist. The paper looks for to find optimal solutions for the generalized problem designing a heuristic. The objective function to diminish in this case is the one of the date of completion of the last work. The designed heuristic was proven in twenty eight different instances.

REFERENCES

- MORTON, Thomas, PENTICO, David. Heuristic Scheduling Systems. Wiley-Interscience, 1993. Pp 241-293.
- PINEDO, Michael, CHAO, Xiuli. Operations Scheduling with Aplications in Manufacturing and Services. Mc Graw Hill Edition, 1999. Pp 20-24
- PINEDO, Michael. Scheduling: Theory, Algorithms and Systems. Prentice-Hall, 2da. Edition, 2001.