Algoritmo heurístico para resolver el problema de programación de operaciones minimizando el tiempo total de ejecución ponderado de las actividades (WJCJ) en un taller de flujo flexible con restricciones de fechas de liberación (RJ), alistamiento (SJK) y máquinas proporcional (QM) en las estaciones

Heuristic algorithm for a flexible flow shop problem minimizing total weighted completion time ($W_J C_J$) with release dates (R_J), setup (S_{JK}) constrains with proportional machines (Q_M) at stations

Luis Ramírez Polo

[1] *Master of Science in Industrial Engineer, Associate Professor,*
Universidad Autónoma del Caribe, Gestión Moderna de Operaciones (GeMop),
luis.ramirez@uac.edu.co

Recibido 30/05/11, Aceptado15/07/2011

ABSTRACT

Scheduling on many stations and machines minimizing total weighted completion time ($w_j C_j$) as objective with released dates and setup's constraints is not a new problem, but it is a low investigated because it has a computational complexity of NP-Hard and in most cases the heuristics are not optimal solutions Our objective in the problem is develop an heuristic to be applied through an algorithm that gives as output a sequence of jobs in each station and machines having different velocities in each one and minimizing the principal objective total weighted completion time.

Keywords: Scheduling; Parallel Machines; Setups; Release dates, heuristic algorithm; flexible flow shop; total weighted completion time.

RESUMEN

La programación de operaciones para un taller donde los trabajos poseen fechas de entrega, alistamiento y ponderación, buscando minimizar el tiempo de terminación ponderado no es un nuevo problema que se a trabajado a nivel investigativo, pero es un problema poco trabajado por ser un problema con una complejidad computacional alta, considerada de tipo NP-HARD, campo donde la mayoría de los casos las heurísticas dan soluciones no *óptimas*. Lo que se muestra en esta investigación es el desarrollo de una heurística que arroje una forma eficiente para programar los trabajos en un taller de m*á*quinas en paralelo que tiene las condiciones antes mencionadas y busque minimizar el tiempo total ponderado en el sistema.

Palabras clave: Programación de operaciones; Máquinas en paralelo; algoritmo heurístico; Taller de flujo flexible; Tiempo total ponderado; tiempo de liberación; alistamientos.

1. INTRODUCTION

The flexible systems are defined as an automatic controlled process that could produce many items between determined ranges [1]. It is a technology that helps to optimize the manufacturing with better times, lower costs and a better quality, through better control systems. The flexible systems are conformed by a group of *k* stations and each station has *mk* parallel machines that could process more than one job, but we need an optimal sequence of the job in each station to guaranty the minimum time of end of each job or an effective flexible system. In 1973 began the first studies of scheduling with flexible flowshops, at that time the objective was to minimize the makespan. Hoogeveen, Schuurman and Woeginger argued the existence of a polynomial approximation to this type of problems which are NP hard. [2]

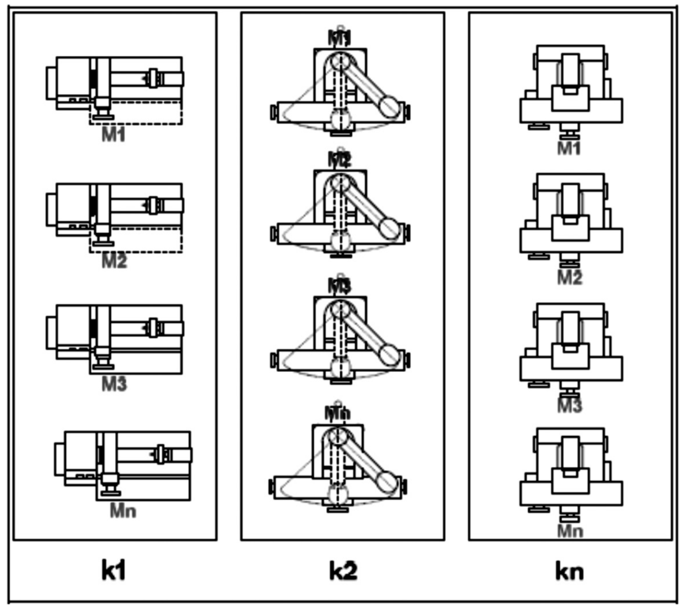
Most studies on scheduling problems assume that machines are available at all times [3]. In the actual configuration of the industry, however, a machine may not always be available in the programming period, due to, for example, a breakdown (stochastic) or preventive maintenance (deterministic), but in this paper considers that the machines are always available. [4]. Our objective in the problem is develop an heuristic to be applied through an algorithm that gives as output a sequence of jobs in each station and machines having different velocities in each one and minimizing the principal objective total weighted completion time that give us solutions to the problem described before to see the sequence, the values of the objectives for each station. The methodology for the solution of the problem in mention will be the application of knowledge of scheduling for easier cases and considering the constraints of our model to make the necessary changes for solving intelligently harder problems[5] [6].

Section 2 presents a description of the problem at hand, section 3 describes the methodology of the heuristic developed in sections 4 and 5 shows the implementation of the heuristics to a computational complexity problem and developed in section 6 shows a comparison of the heuristic developed with other already published and finally the conclusions reached as a result of the investigation.

2. RELATED WORK

In this paper analyze a making decision problem on the scenario, this mean that we have a flexible flowshop with *k* stations that has (M_k) proportional machines ($FFc \mid r_j, S_{jk} \mid 3w_jC_j$). [7] This problem is centered on scheduling with a scenario $FFc \mid r_j, S_{jk} \mid 3w_jC_j$ who has special values and range for the variables as k = [3,5], M_k = [1,3]; j= [15, 20] and S_{jk} for each station, where each job has an independent released date and not all the stages have the same time to pass from a job (j) to a different job (k). The configuration for the problem above is presented below:

Figura 1. Ejemplo de distribución de máquinas; fuente propia.
Figure 1. Facility Layout example; author citation.



This specific problem has some constraints that should have being in mind, for example:

- A job *i* should pass at least one time for each station.
- Each job *i* has a termination time that depends for the first station on the processing time of each job on each machine of each station and the previous one, for other stations is maximum between the termination time in first station and the previous job on the machine plus the processing time.

3. HEURISTIC DEVELOPMENT

Step 1: The bottleneck station is determined calculating the sum of processing times of all jobs and multiply the results to the sum of machines velocities. Then, the value obtained is divided on the number of machines. This operation is done for each station and the bottleneck stage is the one with the highest value.

Step 2: The first job is scheduled having in mind the minimum released date (r_j), if there are ties, they are broken by the highest w_j value and if theties persist it is indifferent the selection. This job is scheduled in the fastest machine.

Step 3: The sum of r_j, S_{jk} and p_j is calculated for each following jobs on each proportional machine of the stage, and the job is scheduled in the machine where the minimum value was gotten. If there are ties the job with the highest w_j value is selected. This step finish when all the jobs are scheduled in stage. Then go back to step 2 only if the next stage is not the bottleneck station. Otherwise go on step 4.

Step 4: The bottleneck is scheduled using the methodology of step 3, the next station are assigned with the same sequence of the bottleneck station.

Step 5: At the same time it is applied the step 3 to be compared with bottleneck sequence, the scheduled with the minimum C_{max} is chosen.

Step 6: It is calculated the makespan of the system and total weighted completion time which is the objective to minimize.

4. APPLICATION OF THE HEURISTIC FOR ONE INSTANCE

For this case we have: Flexible flow shop problem with r_j and S_{jk} and objective to minimizing the w_jC_j, with five jobs, and 3 stations with different number of machines in every station.

Figura 2. Ejemplo de una instancia; fuente propia
Figure 2. Example for a instance; Author citation

Number of jobs	5
Number of stations	3

Number of machines by stations		
S1	*S2*	*S3*
2	3	2

Standar velocity by machine		
S1	*S2*	*S3*
M1 = 2	M1 = 3	M1 = 2
M2 = 1	M2 = 1	M2 = 1
	M3 = 2	

Data

Job	*1*	*2*	*3*	*4*	*5*
P_1	10	5	6	7	10
P_2	2	8	8	3	2
P_3	3	10	4	4	9
W_j	3	4	8	10	1
r_j	0	2	0	2	1

S_{jk}

S1	1	2	3	4	5
1	-	8	9	6	8
2	11	-	5	7	6
3	4	5	-	4	9
4	6	2	8	-	9
5	2	7	6	9	-

S3	1	2	3	4	5
1	-	10	7	3	7
2	4	-	8	11	7
3	5	7	-	6	10
4	10	7	11	-	10
5	9	3	10	7	-

S2	1	2	3	4	5
1	-	10	3	10	7
2	10	-	11	9	11
3	7	5	-	9	7
4	9	8	3	-	11
5	8	5	5	8	-

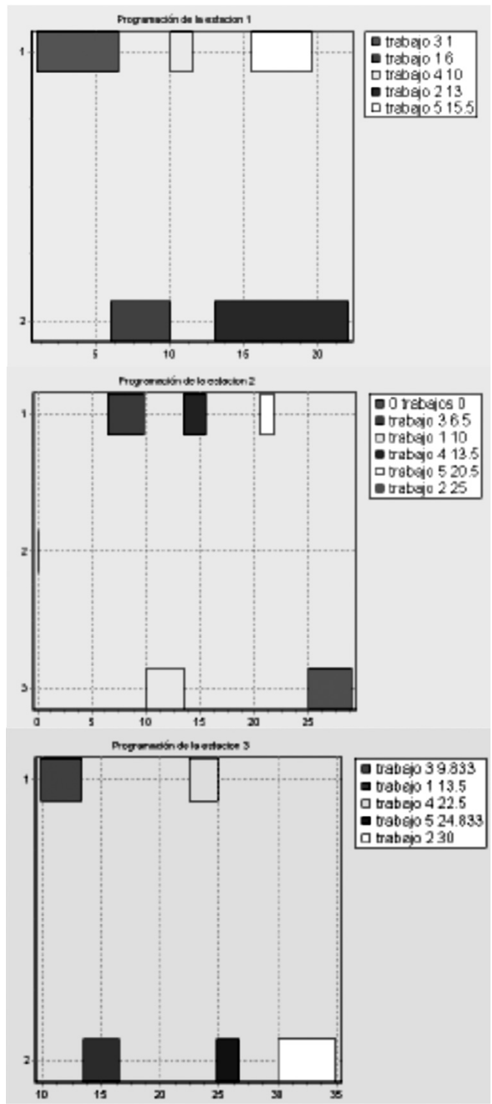
The solution of this problem through the application of our heuristic gives us the next outputs.

Figura 3. Programación en las estaciones; fuente propia
Figure 3. Schedule on the stations; author citation

Station 1 Schedule

Station 2 Schedule

Station 3 Schedule



$C(1) = 39$ $w_1 = 3$ $w_jc_j(1) = 117$
$C(2) = 52$ $w_2 = 4$ $w_jc_j(2) = 208$
$C(3) = 29$ $w_3 = 8$ $w_jc_j(3) = 232$
$C(4) = 38$ $w_4 = 10$ $w_jc_j(4) = 380$
$C(5) = 44$ $w_5 = 1$ $w_jc_j(5) = 44$
$Cmax = 52$ $\sum w_jc_j = 981$

On this results it was applied the step of compare the schedule of bottleneck station with the sum of r_j, s_{jk} and P_j to get a better distribution on the machines and minimize the total weighted completion time (w_jC_j).

5. COMPUTATIONAL COMPLEXITY

For this program the complexity obtained is O(n*m*t), associated to tree types of variables.

n: Number of stages

m: Maximum number obtained number of machines in an station

t: Number of jobs

6. EXPERIMENTAL RESULTS

The developed heuristic was run with 22 cases at random, each of the different cases differ in the release dates, processing times, stations, machine speeds, weight, number of stations, the machines by stations. To evaluate the performance of this heuristic was evaluated with two different heuristics have posted the same problem # 1 [8], 2 [9],which yielded the following results: The first table shows how many times each heuristic reached the first, second and third place respectively in relation to the tests conducted in the second table shows a proportion of the time which appears more heuristic in that position and the third shows a relation of the result of the makespan with respect to the best result obtained.

Tabla 1. Posiciones de las diferentes heurísticas; fuente propia
Table 1. Ranking with tree different heuristic; author citation

	Ranking		
	1st	**2nd**	**3rd**
#1	15	3	4
Us	3	12	7
#2	4	7	11

This also could be represented by percentage as in the follow table.

Tabla 2. Posiciones porcentuales para cada una de las heurísticas; fuente propia
Table 2. Percentage positions for each of the heuristics; author citation

	Ranking		
	1st	**2nd**	**3rd**
#1	0,68	0,14	0,18
Us	0,14	0,55	0,32
#2	0,18	0,32	0,5

Also it was calculated the differences between the solution of each group and the best one for all analyzed cases divided on the best result

Despite the heuristic solutions developed only won three times, the last table shows that we have the minimum value of the worst. This means that our worst result is not far from the best result, in contrast with the heuristic which won more cases with a minimum target value raised in the occasions that you do not get a good result is far from it. This result by applying to the reality that the heuristic could be said plated in this publication is better, on many occasions would prefer to choose a model that does not necessarily always gave me the best if not to be sure that when he gave me a bad outcome is not away the best. [10] [11]

Tabla 3. Diferencias obtenidas con respecto a los mejores resultados para cada heurística; fuente propia
Table 3. Differences obtained with respect to the best outcome for each of the heuristics; author citation

	Ranking		
	Us	**#1**	**#2**
1	1,68	0,00	2,74
2	0,30	0,00	0,60
3	2,47	0,00	2,37
4	0,00	0,15	0,01
5	0,07	0,00	0,17
6	0,00	0,32	0,35
7	1,33	0,00	1,67
8	0,35	0,00	0,31
9	1,92	6,48	0,00
10	0,89	0,00	0,77
11	0,10	0,54	0,00
12	0,97	0,00	0,40
13	1,42	0,00	0,21
14	1,52	0,00	1,58
15	0,63	0,00	0,14
16	0,38	0,08	0,00
17	0,18	0,39	0,00
18	0,00	0,20	0,26
19	1,03	0,00	1,16
20	0,09	0,00	0,60
21	0,71	0,00	1,47
22	0,27	0,00	0,40

7. CONCLUSIONS

We note that the analyzed scenario is a NP-hard problem that has no answer or a solution for all the examples [12] [13] , cases, because this is in an area with no possibility of fair comparisons and find the effectiveness of the problem. [14] Although the objective of minimizing the wjCj, release dates are strong forces to consider, because if the job is not available, it is possible to make the process at that stage and in the following. It can be concluded while the bottleneck station is an important starting point for scheduling flexible flow shops, as it depends on the programming of the following works and is the most restricted to minimize the completion time of work, which is one of the most searched in the shops that handle the distribution [15]. As a result of this study suggest that a methodology is not always offered to me in many cases the best result is the best decision to make without measuring the wrong case, the ideal is to combine this with the question: ¿What good is when there gives me the best option? It would be best is to reach a very good result, but if you get this would not be too bad.

At the present time the closer we come to processes of iteration and the search for optimal solutions faster due to technological advances, while assessing the proper way to obtain this solution is something that will govern in the literature.

References

[1] Pinedo, Michael. Scheduling: Theory, Algorithms and Systems. Prentice-Hall, 2da. Edición (2001)

[2] J.A. HOOGEVEEN, P. SCHUURMAN, AND G.J. WOEGINGE, Non approximability results for scheduling problems with minsum criteria, Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization (IPCO), 353–366, 1998

[3] Pinedo, Michael, Xiuli Chao. Operations Scheduling with applications in Manufacturing and Services. McGraw-Hill/Irwin; Bk&Disk edition (1998)

[4] Xie. J, Wang X. Complexity and Algorithms for Two-Stage Flexible Flowshop Scheduling with Availability Constraints, Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China.

[5] T'kindt Vincent and Billauat Jean-Charles, Multicriteria Scheduling Theory, Models and Algorithms, Ed Stringer , 2002.

[6] J. X. Xie and X. J. Wang. Complexity & algorithms for two-stage flexible flowshop scheduling with availability constraints. Computers & Mathematics with Applications, 50(10-12):1629–1638, 2005.

[7] Sipper Daniel, Bulfin Robert. Planeación y control de la Producción. McGraw-Hill, 1998

[8] Minimization of Cmax in Parallel Machines with Identical Parallel Machines , Release and Setup Times. Celia Aleán Zapata, Vanesa Manotas Niño, María Orozco Arredondo, Latin American and Caribbean Consortium of Engineering Institutions, LACCEI 2006

[9] An Heuristic Alternative to Optimize Makespan over M Parallel Machines with Release and Sequence-Dependent Setup Times Pm | rj, sjk | Cmax Julián Lopez, Viviana Rozo, Iván Saavedra, Latin American and Caribbean Consortium of Engineering Institutions, LACCEI 2006

[10] J. X. Xie,W. X. Xing, Z. X. Liu, and J. F. Dong. Minimum deviation algorithm for two-stage no-wait. Flowshops with parallel machines. Computers & Mathematics with Applications, 1857–1863, 2004.

[11] H. Xuan and L. Tang. Scheduling a hybrid flowshop with batch production at the last stage. Computers & Operations Research, 34(9):2718–2733, 2007.

[12] Lopez De Haro, Santiago, P. Sanches Martín y Javier Conde. Secuenciación de tareas mediante metaheurísticos. VIII Congreso de ingeniería de organización leganés.(2003)

[13] W. Xiao, P. Hao, S. Zhang, and X. Xu. Hybrid flow shop scheduling genetic algorithms. In Proceedings of the 3rd World Congress on Intelligent Control and Automation, pages 537–541. IEEE press, 2000.

[14] V. Yaurima, L. Burtseva, and A. Tchernykh. Hybrid flowshop with unrelated machines, sequencedependent setup time, availability constraints and limited buffers. In press at Computers & Industrial Engineering, 2009.

[15] K. C. Ying. An iterated greedy heuristic for multistage hybrid flowshop scheduling problems with multiprocessor tasks. In press at Journal of the Operational Research Society, 2009.