Comparison of Approximation Algorithms for Unrelated Parallel Machines to Minimize the Weighted Makespan

[1]Tariq S.Abdul-Razaq and [2]Najwa Raheem Mustafa
[1]Dept.of Mathematics,College of Science, Al-Mustansiriyah University
[2]Dept. of Mathematics, College of Science for Women, Baghdad University

الخلاصة

تمت دراسة جدولة المكائن المتوازية غير المترابطة.في هذه الحالة يجب جدولة مجموعة n من الاعمال على m من المكائن المتوازية غير المترابطة . كل عمل متوفر للمعالجة عند الزمن صفر وكل ماكنة يمكنها معالجة عمل واحد على الاكثر في نفس الوقت وكل عمل يمكن إنجازه بواسطة ماكنة واحدة في نفس الوقت . تمت دراسة حالة تطبيقية لجدولة اعمال في ورشة قطع لتصغير زمن الانهاء الاعظم الموزون . تم اقتراح خمسة خوارزميات ودراسة ادائها.

ABSTRACT

The problem of scheduling of unrelated parallel machines is considered. In this environment, a set of n jobs has to be scheduled on m unrelated parallel machines. Each job is available for processing at time zero and each machine can process at most one job at a time and a job can be processed by at most one machine at a time. A case study is considered to schedule jobs in a cutting workshop to minimize the weighted makespan. Five algorithms are proposed and their performance is studied.

INTRODUCTION

The problem considered in this paper is the scheduling of unrelated parallel machines. In parallel machines environment there are multiple machines. In identical parallel machines environment all machines operate at the same speed, while in uniform parallel machines environment each machine has its own speed. For unrelated parallel machines; there are multiple machines with different job-related speeds, that is the processing times are unrelated. In this research environment, a set N of n jobs, 1,…, n, each of which has to be scheduled on one of m machines, $M_1,...,M_m$ is given. Each job j has a processing requirement p_j ,weight w_j and is available for processing at time zero. Each machine can process at most one job at a time and a job can be processed by at most one machine at a time. All machines start working at time zero and process their jobs sequentially. For the unrelated parallel machines, the speed of machine M_i on job j, v_{ij}, depends on both the machine and the job; job j requires p_j / v_{ij} processing time on a machine M_i. We define $p_{ij} = p_j / v_{ij}$ (1). The notation N^i denotes the set of jobs assigned to machine M_i. Let C_j denotes the completion time of job j and CM_i denotes the completion time of the last job on machine M_i, that is:

$$CM_i = \sum_{j \in N^i} p_{ij} \qquad i=1,\dots, m, j=1,\dots,n$$

The maximum completion time (makespan) C_{max} is defined by (2):

$C_{max} = \max\{CM_1,...,CM_m\}$, we can define C^w_{max} by:

$C^w_{max} = \max\{CM_1,...,CM_m\}$ with respect to some priority rules that depend on job weight; the objective is to minimize the weighted makespan C^w_{max}.

There is extensive literature describing approximation algorithms for unrelated parallel machine scheduling problems. Horwitz and Sahni (1976) proposed several exact and approximate algorithms for some special cases of the $R//C_{max}$ problem. Also, Ibarra and Kim (3), Davis and Jaffe (4) proposed several algorithms to solve the problem $R//C_{max}$.There are many papers that present an experimental comparison of approximation algorithms, some of which are based on linear programming and others based on local search heuristics. For example, Hariri and Potts (5) solve the $R//C_{max}$ problem. The Branch and bound method can also be used, as was the case with Salem, Anagnostopoulos and Rabadi [2] when solving the problem $R//C_{max}$.

Bruno, Coffman and Sethi (1974) and Lenstra et al.(1977) (6)showed that the problem of minimizing total weighted completion time on two identical parallel machines is NP-hard, thus $R//\sum w_j C_j$ is NP-hard in the strong sense (7).

If there is only one machine the problem is solved to optimality by ordering the jobs in a non-decreasing order of the ratio p_j / w_j (the SWPT rule) (Smith 1956). Hence, the problem reduces assigning the jobs appropriately to the machines and then sequencing the jobs on each machine by the SWPT rule.

Karp (1972) show that the problem of scheduling two identical parallel machines to minimize the maximum completion time is NP-hard. Clearly, the more general unrelated parallel machine problem is also NP-hard (5). Thus for the $R//C^w_{max}$ problem it seems unlikely that a polynomial time algorithm that always produce optimal solution exists.

The Problem $\boldsymbol{R//C^w_{max}}$

The problem $R//C^w_{max}$ can be formulated as a linear program as follows:

Minimize $Z = C^w_{max}$.

Subject to

$$\sum_{j=1}^{n} x_{ij} w_j p_{ij} \le C^w_{max} \qquad i = 1,...,m$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad j = 1,...,n$$

$x_{ij} \in \{0,1\}$ i=1,…,m;j=1,…,n

Where x_{ij} is an assignment variable that is equal to 1 if job j is assigned to machine M_i and 0 otherwise.

The best time of the jth job, denoted by b_j, equals $\min_{1 \le i \le m} p_{ij}$. The efficiency of the ith machine on the jth job, denoted by ef_{ij}, equals to b_j / p_{ij} (4). The maximum efficiency is one. Using these concepts, the following algorithm is a modification of the algorithm proposed by Davis and Jaffe (4).

Algorithm 1: Algorithm WCM1

Step (1): *$N = \{1,\ldots,n\}, N^i = \phi$. For j=1,…, n , find $b_j = \min_{1 \le i \le m} p_{ij}$;*

for j=1,…,n;i=1,…,m,find $ef_{ij} = b_j / p_{ij}$;

for i=1,…,m , create a list of the jobs j=1,…,n sorted in a non-increasing order of $w_j ef_{ij}$. Set $sum_i = 0$, $CM_i = 0$ for i=1,…,m.

Designate all machines as 'available 'and all jobs as 'unassigned';

Step (2): *If $N = \phi$, go to step(6),else find a machine i such that sum_i is minimal among all available machines;*

Step (3): *Find the next unassigned job j on i's list;*

Step (4): *If j does not exist or if $ef_{ij} < 1/\sqrt{m}$ then mark i as unavailable;*

Step (5): *Otherwise assign job j to machine i ; $N = N - \{j\}$, $N^i = N^i \cup \{j\}$. Set $sum_i = sum_i + p_{ij}$, $CM_i = CM_i + p_{ij}$, return to step(2).*

Step (6): *$C^w_{max} = \max_{1 \le i \le m} \{CM_i\}$.*

Figure -1: The WCM1 algorithm

The ratio ef_{ij} can be used to propose another algorithm that is called WCM2.

Algorithm 2: Algorithm WCM2

Algorithm WCM2 is similar to algorithm WCM1 except that in step (1) the jobs are sorted in a non-decreasing order of ef_{ij} / w_j.

Figure -2: The WCM2 algorithm

Ibarra and Kim (3) proposed several algorithms for the problem $R//C_{max}$. Some of these algorithms are modified to fit the problem $R//C_{max}^{w}$ as follows.

Algorithm 3: Algorithm WCM3

Step (1): $N=\{1,...,n\}$, $N^i=\phi$ *and* $sum_i=0$ *for i=1,…, m;*

Step (2): *If* $N=\phi$*,go to step (4);*

Step (3): *Otherwise find a job* $j \in N$ *such that* $\min_{1\le i\le m}\{sum_i+p_{ij}\} \le \min_{1\le i\le m}\{sum_i+p_{ij'}\}$ *for all* $j' \in N$*; let i be such that* sum_i+p_{ij} *is minimum;* $sum_i=sum_i+w_j p_{ij}$, $CM_i=CM_i+p_{ij}$, $N^i=N^i\cup\{j\}$, $N=N-\{j\}$, *return to step (2);*

Step (4): $C_{max}^{w}=\max_{1\le i\le m}\{CM_i\}$.

Figure 3: The WCM3 algorithm

Other algorithms can be proposed by first arranging jobs according to some priority rules as in the following two algorithms.

Algorithm 4: Algorithm WCM4

Step (1): $N=\{1,...,n\}, N^i=\phi, sum_i=0$, *i=1,…,m;*

Step (2): *For each job j , find* $p'_{min}(j)=\min_{1\le i\le m} p_{ij}/w_j$

Step (3): *Order the jobs in* N *according to non-decreasing order of* $p'_{min}(j)$*;*

Step (4): *If* $N=\phi$*, go to step(6);*

Step(5): *Else , find the machine i such that* $sum_i+w_j p_{ij} \le sum_k+w_j p_{kj}$ *for all* $k=1,...,m$ *;set* $sum_i=sum_i+w_j p_{ij}$ *;* $N=N-\{j\}$, $CM_i=CM_i+p_{ij}$ *;* $N^i=N^i\cup\{j\}$*;return to step(4);*

Step (6): $C_{max}^{w}=\max_{1\le i\le m}\{CM_i\}$.

Figure-4: The WCM4 algorithm

Also the following algorithm is a modification of the LRPT-FM (Longest Remaining Processing Time on the Fastest Machine) rule.

Algorithm 5: Algorithm WCM5

Step (1): $N = \{1,...,n\}, N^i = \phi, sum_i = 0$, $i=1,\dots,m;$

Step (2): *For each job j , find* $p'_{max}(j) = \max_{1 \le i \le m} p_{ij} / w_j$

Step (3): *Order the jobs in* N *according to non-increasing order of* $p'_{max}(j)$;

Step (4): *If* $N = \phi$, *go to step(6);*

Step (5):*Else , find the machine i such that* $sum_i + p_{ij} \le sum_k + p_{kj}$ *for all k=1,…,m ; set* $sum_i = sum_i + p_{ij}; CM_i = CM_i + p_{ij}$,

$N = N - \{j\}; N^i = N^i \cup \{j\}$;*return to step(4);*

Step (6): $C^w_{max} = \max_{1 \le i \le m}\{CM_i\}$.

Figure -5: The WCM5 algorithm

Case Study

The Five algorithms : WCM1,WCM2, WCM3,WCM4 and WCM5 are applied in the cutting workshop in Al-Karama General State Company. The company has a complete engineering department for design and technology that works jointly with the planning and follow-up department and different factories to accomplish production operations within the annual plan, in addition to some specified orders for special projects. One of the divisions of the planning and follow-up department is the cutting workshop, which carries out a huge and basic part in preparing and providing production work orders the raw materials for all factories in primary measurements specified by technological procedure of these parts production.

The work of the cutting workshop is of great importance in preparing and providing raw materials in correct dimensions and required quantities within accurate times and specified types to all factories by best utilization of available self capabilities. Thus the work of the cutting workshop is a bottleneck to progress of production operation in factories and the company.

The cutting workshop environment includes the following components:

a) Machines: There are different types of machines used in this unit. The speed of each machine depends on its own specifications and the raw materials.

b) Raw materials: There is a wide range of metal ores and materials used in the workshop works which are divided into five kinds according to their cutting speed, these types are: Aluminium, Stainless steel, Other types of steel , Teflon and different plastics and Cooper and brass.

c) Work Style: The cutting workshop receives work orders to prepare materials weekly. The company runs an annual production plan and the planning department has a monthly plan for the factories production and the cutting workshop prepares materials at least one month ahead.
d) Constraints: Jobs in the cutting workshop include cutting shafts, blocks and plates ,each of these material need certain time as a loading cost. Some machines have certain constraints, for example, they cannot handle some raw materials. Raw materials are available in standard measurements.

The algorithms WCM1, WCM2, WCM3, WCM4 and WCM5 are implemented on a case study of 10 assemblies. These assemblies consist of different numbers of jobs and machines. The jobs vary in their types and raw materials, also the machines are of different types. The efficiency of the proposed algorithms is tested using programs coded in Microsoft FORTRAN Power Station version 4.0; the sketches were drawn using MATLAB 6.5. Both codes are executed on a Pentium III 1GHz personal computer with 256 MB memory.

The schedules yielding from these algorithms are compared. Table (1) presents the results obtained by these algorithms and the results are compared to the best of them using the percentage relative deviation from the best value (PRD), calculated as $\frac{H-B}{B}*100$, where H and B represents the heuristic and best values, respectively. Also, they are compared using the deviation (DEV) of each value w.r.t. arithmetic mean of all values. The best result is presented in bold.

Table -1: Comparison of the results for C^{w}_{max}

ASSEMBLY	NO. OF JOBS	NO.OF MACHINES	ALGORITHM	VALUE	PRD	DEV
1	21	6	WCM1	750	79.42583	165.6
			WCM2	522	24.88038	-62.4
			WCM3	733	75.35885	148.6
			WCM4	499	19.37799	-85.4
			WCM5	418	0.000	-
2	18	6	WCM1	836	12.66846	32.8
			WCM2	742	0.000	-61.2
			WCM3	865	16.57682	61.8
			WCM4	811	9.29919	7.8
			WCM5	762	2.69542	-41.2

3	16	5	WCM1	1781	69.1358	363.8
			WCM2	1312	24.59639	-
			WCM3	1239	17.66382	-
			WCM4	1701	61.53846	283.8
			WCM5	1053	0.000	-
4	22	5	WCM1	1546	0.000	0.000
			WCM2	1546	0.000	0.000
			WCM3	1546	0.000	0.000
			WCM4	1546	0.000	0.000
			WCM5	1546	0.000	0.000
5	20	7	WCM1	1138	37.10843	118
			WCM2	1118	34.6988	98
			WCM3	995	19.87952	-25
			WCM4	1019	22.77108	-1
			WCM5	830	0.000	-190
6	16	6	WCM1	962	55.16129	133.6
			WCM2	895	44.35484	66.6
			WCM3	783	23.06452	-65.4
			WCM4	902	45.48387	73.6
			WCM5	620	0.000	-
7	14	5	WCM1	481	37.03704	-40.8
			WCM2	481	37.03704	-40.8
			WCM3	815	132.1937	293.2
			WCM4	481	37.03704	-40.8
			WCM5	351	0.000	-
8	15	5	WCM1	1786	6.05701	17
			WCM2	1786	6.05701	17
			WCM3	1753	4.09739	-16
			WCM4	1836	9.02613	67
			WCM5	1684	0.000	-85
9	17	6	WCM1	1130	9.07336	-82.8
			WCM2	1378	33.01158	165
			WCM3	1278	23.35907	65
			WCM4	1242	19.88417	29
			WCM5	1036	0.000	-
10	19	5	WCM1	1665	25.18797	95.2
			WCM2	1665	25.18797	95.2
			WCM3	1665	25.18797	95.2
			WCM4	1524	14.58647	-45.8
			WCM5	1330	0.000	-

It can be seen from table (1) that algorithm WCM5 is the best one. It generates the best solution almost for all the assemblies given (nine out of 10).

The performance of other algorithms varies radically from an assembly to another. It is interesting to see that the five algorithms perform identically on assembly (4). Except algorithm WCM5, other algorithms perform badly on some assemblies and well on others. The running time for the program of 10 assemblies is 1.500000E – 01 seconds. The results are presented in the figure (6).

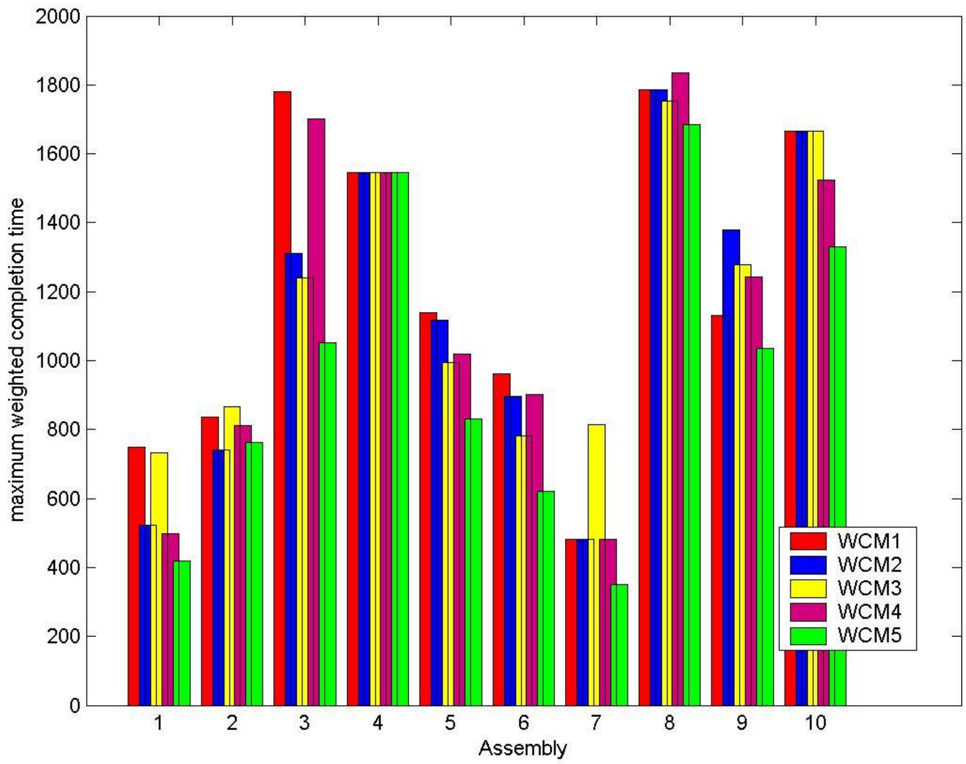

Figure-6:The performance of algorithms WCM1, WCM2, WCM3, WCM4 and WCM5

Conclusions

In this work an applied problem was studied, which is the problem of scheduling n jobs on m unrelated parallel machines.Several algorithms are proposed to minimize the maximum weighted completion time. They are applied on the work of a cutting workshop in Al-Karama general state company and their performance is analyzed hoping to suit the company demands. Algorithm WCM5 is the best one for the maximum weighted completion time problem of unrelated parallel machines. The performance of other algorithms for the problems (minimizing the maximum weighted completion time) varies between an assembly and another one. To improve the performance, one can study the preemption of jobs. Also, on-line scheduling, i.e. scheduling when jobs arrive over time, can be studied.

REFERENCES

1. Karger, D., Stein, C. and Wein, J., Scheduling Algorithms, in Atallah, M.J.(ed.) CRC Handbook on Algorithms and Theory of Computation, CRC Press, (1998).
2. Salem,A., Anagnostopoulos, G.C. and Rabadi, G, A Branch-and-Bound Algorithm for Parallel Machine Scheduling Problems, in Proceedings of Harbour, Maritime & Multimodal Logistics Modeling and Simulation Workshop, a Publication of the Society for Computer Simulation International (SCS), Petrofina, Italy:88-93, October (2000).
3. Ibarra, O.H. and Kim, C.E., Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors, Journal of the Association for Computing Machinery 24:280-289, (1977).
4. Davis,E.& Jaffe, J.M., Algorithms for Scheduling Tasks on Unrelated Processors, Journal of the Association for Computing Machinery 28:721-736 (1981).
5. Hariri, A.M.A. and Potts, C.N., Heuristics for Scheduling Unrelated Parallel Machines, Computers and Operations Research 18:323-331, (1991).
6. Lenstra, J.K., Rinnooy Kan, A.H.G. & Brucker,P.,Complexity of Machine Scheduling Problems, Annals of Discrete Mathematics 1:343-362(1977).
7. Valente, J.M.S. and Alves, R.A.F.S., Improved Heuristics for the Early/Tardy Scheduling Problem with no Idle Time, Computers & Operations Research 32 :557-569(2005).