The Journal of
Mathematics and Computer Science

Available online at
http://www.TJMCS.com

The Journal of Mathematics and Computer Science Vol .1 No.4 (2010) 349-354

A fuzzy heuristic algorithm for the flow shop scheduling problem

Mehdi Heydari

Mheydari@iust.ac.ir

Emran Mohammadi

E_mohammadi@iust.ac.ir

Iran university of science and technology, department of industrial engineering

Received: July 2010, Revised: October 2010
Online Publication: December 2010

Abstract

The two-machine flow shop problem with the objective of minimizing makespan that is known as Johnson problem is now standard fundamental in the theory of scheduling. We generalize Johnson's results for more than two machine problems, using a fuzzy heuristic algorithm. Performance of the new algorithm is analyzed with some numerical examples. To evaluate the performance of the proposed heuristic, we have used it on some small size problems and the results are compared with optimum scheduling. Notice that scheduling the problems with large sizes, is NP hard.

Key words

Scheduling; Fuzzy heuristic algorithm; Membership function; Flow shop

1. Introduction

In this section we first define the necessary definition for modeling the flow shop preemptive problem, subject to minimizing makespan and then Johnson’s rule is investigated briefly.

A flow shop consists of n machines that are continuously available and organized as a production line. Each job has n operations that have to process sequentially on the machines, for example, the second operation of any job cannot start before completion the first operation. In addition, there is a virtual $FIFO$ for the subsequent machines. A job J is specified by its processing times on machine $M_k, k=1,\ldots,n$ that are shown respectively with $t_{jk}, k=1,\ldots,n$. In addition, each machine can handle only one job at a time. Completion date for job *j* is denoted by C_j. Our objective is to find a schedule to minimize the makespan $(C_{\max})$.

The classical two-machine flow shop problem with the objective of minimizing makespan is also known as Johnson problem [1] and the problem becomes NP-complete for three and more machines [2].The results originally obtained by Johnson are now standard fundamentals in the theory of scheduling. Since Johnson's seminal work has been one of the most extensively studied topics in the literature on production planning [3]. For two-machine flow shop problem, first machine can process all jobs continuously with no idle time (see Fig.1).

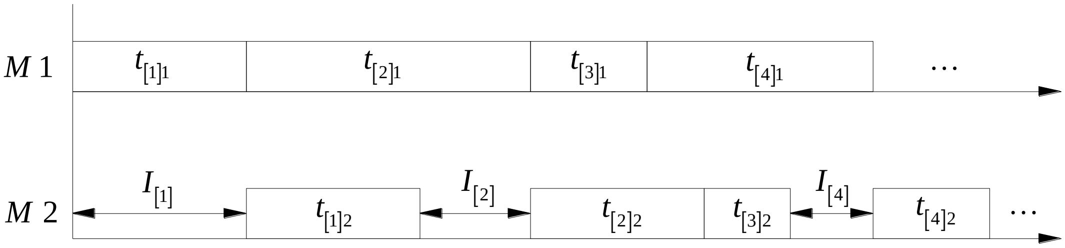


Fig1: Two machine flow shop problem

The idle time of the second machine before the processing of job j , is denoted by $I_{[j]}$ and the makespan can be calculated as follows:

$$C_{\max} = \sum_{j=1}^{n} t_{j2} + \sum_{j=1}^{n} I_j \qquad (2)$$

Note that the makespan is the maximum completion time of all jobs (the end of the second task of the last job). Since the first term in (2) is a constant, an equivalent problem is to minimize the second term. Observe that:

$$I_{[1]} = t_{[1]1}$$

$$I_{[2]} = \max\{0, t_{[1]1} + t_{[2]1} - t_{[1]2} - I_{[1]}\}$$

$$I_{[3]} = \max\{0, t_{[1]1} + t_{[2]1} + t_{[3]1} - t_{[1]2} \ - t_{[2]2} - I_{[1]} - I_{[2]}\}$$

In general:

$$I_{[j]} = \max\left\{0, \sum_{i=1}^{j} t_{[i]1} - \sum_{i=1}^{j-1} t_{[i]2} - \sum_{i=1}^{j-1} I_{[i]}\right\}$$

Also, observe that:

$$I_{[1]} + I_{[2]} = \max\left\{t_{[1]1}, t_{[1]1} + t_{[2]1} - t_{[1]2}\right\}$$

$$I_{[1]}+I_{[2]}+I_{[3]}=\max\{t_{[1]1},t_{[1]1}+t_{[2]1}-t_{[1]2},\ t_{[1]1}+t_{[2]1}+t_{[3]1}-t_{[1]2}-t_{[2]2}\}$$

In general:

$$\sum_{i=1}^{j} I_{[i]}=\max_{1\le k\le j}\left\{\sum_{i=1}^{k} t_{[i]1}-\sum_{i=1}^{k-1} t_{[i]2}\right\}$$

For convenience, let:

$$Y_k=\sum_{i=1}^{k} t_{[i]1}-\sum_{i=1}^{k-1} t_{[i]2}$$

So that;

$$\sum_{i=1}^{j} I_{[i]}=\max_{1\le k\le j}\{Y_k\}$$

Moreover, the desired objective is to minimize;

$$\sum_{i=1}^{j} I_{[i]}=\max_{1\le k\le j}\{Y_k\}$$

It is well known that in the offline model, preemption is redundant and the above problem is solved by Johnson's rule [13]. According to Johnson's rule, an optimal sequence can be characterized by the following lemma for ordering pairs of jobs.

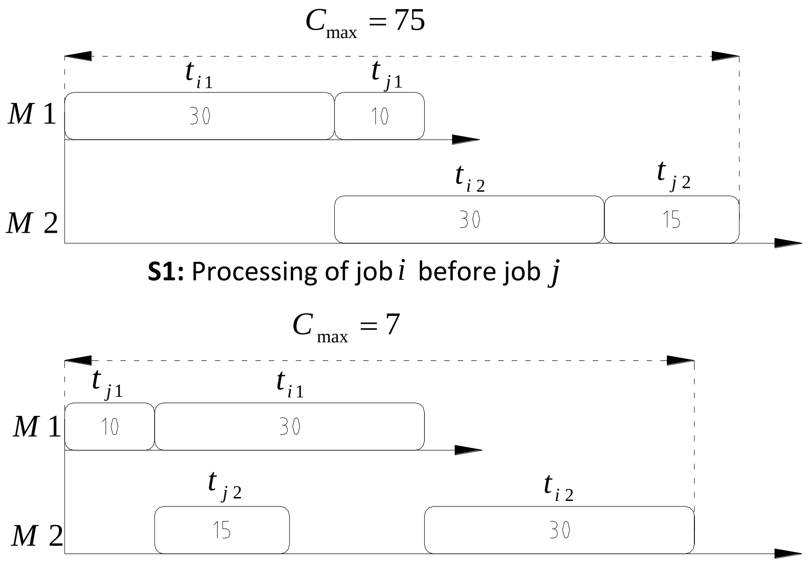
lemma1-(Johnson's rule): Job i precedes job j in an optimal sequence if:

$$\min\{t_{i1},t_{j2}\}\le\min\{t_{i2},t_{j1}\} \qquad (3)$$

For more illumination consider two job i and j with following specifications (table-1). According to (3), $\min\{t_{i1}=30,t_{j2}=15\}\not\le\min\{t_{i2}=30,t_{j1}=10\}$ then job j is preferred to job i (see Fig.2) .

Table 1: Job's specifications

x	t_{x1}	t_{x2}
i	30	30
j	10	15



S1: Processing of job i before job j

S1: Processing of job j before job i

Fig.2: An example of Johnson's rule

Johnson's rule yields a transitive ordering among the jobs, and every job sequence satisfying (3) has the minimum makespan. A practical algorithm according to Johnson's rule is as follows:

Step1- Specify set $U = \left\{ J \,\middle|\, t_{j1} < t_{j2} \right\}$ and set $V = \left\{ J \,\middle|\, t_{j1} \geq t_{j2} \right\}$.

Step2- Arrange members of U increasingly according to t_{j1} and arrange members of V decreasingly according to t_{j2}.

Step3- A sequence of arranged U and V is the optimum schedule.

For the makespan criterion and more than two machine problems $(m > 2)$, it is difficult to generalize the above results. However, the extensions of Johnson's rule for $m = 3$, present some new guidelines. In his original presentation, Johnson showed that a generalization is possible when the second machine is dominated (i.e., when no bottleneck could possibly occur on the second machine).

Lemma2-(extension of Johnson's rule):

1- If $\min_k \{t_{k1}\} \geq \max_k \{t_{k2}\}$, then job i precedes job j in an optimal schedule if :

$$\min\left\{t_{i1}+t_{i2}, t_{j2}+t_{j3}\right\} \leq \min\left\{t_{i2}+t_{i3}, t_{j1}+t_{j2}\right\}$$

2- If $\min_k \{t_{k3}\} \geq \max_k \{t_{k2}\}$, then job i precedes job j in an optimal schedule if :

$$\min\left\{t_{i1}+t_{i2}, t_{j2}+t_{j3}\right\} \leq \min\left\{t_{i2}+t_{i3}, t_{j1}+t_{j2}\right\}$$

Using of the above lemma we present a new fuzzy heuristic algorithm.

2. A fuzzy heuristic algorithm (H)

According to Johnson's rule all jobs are associated to one of two set $U=\left\{ j \middle| t_{j1} < t_{j2} \right\}$ or set $V=\left\{ j \middle| t_{j1} \geq t_{j2} \right\}$, in the other hand, jobs processing time on the two machine problem will be changed increasingly(U) or decreasingly(V). Here we want to extend Johnson's rule for more than two machines. In our new algorithm jobs can be either in fuzzy set $\tilde{U}$ with membership function $\mu_{\tilde{U}}(J)$ and fuzzy set $\tilde{V}$ with membership function $\mu_{\tilde{V}}(J)$. Proposed algorithm is as follows:

Step1- For each job J calculate $S_{j1}=\sum_{k=1}^{a} t_{jk}$ and $S_{j2}=\sum_{k=\left[\frac{n}{2}\right]+1}^{n} t_{jk}$, $a=\left[\frac{n}{2}\right]+\left(\frac{\frac{n}{2}-\left[\frac{n}{2}\right]}{\frac{n}{2}-\left[\frac{n}{2}\right]+\varepsilon}\right)$,

where is a very small positive number.

Step2- Specify $\mu_{\tilde{U}}(J)=\frac{S_{j1}}{S_{j1}+S_{j2}}$ and $\mu_{\tilde{V}}(J)=\frac{S_{j2}}{S_{j1}+S_{j2}}$.

Step3- Arrange members of U decreasingly according $\mu_{\tilde{U}}(J)$ to t_{j1} or arrange members of V increasingly according to $\mu_{\tilde{V}}(J)$.

Step4- A sequence of arranged U or V will be an acceptable schedule.

3. Numerical example

For more illumination Suppose there are 4 jobs with the following specifications in the shop according to table-2. Each job must be processed on the 5 machine in flow shop environment.

Table 2: Job's specifications

J	t_{j1}	t_{j2}	t_{j3}	t_{j4}	t_{j5}
1	5	2	3	6	3
2	1	4	3	4	4
3	5	4	2	4	4
4	3	6	3	5	5

Implementation of the new algorithm is depicted in table-3. Notice that because $n=5$ than we have $\left[\frac{n}{2}\right]+1=3$ and accordingly $a=3$.

Table 3: Calculating parameters of the algorithm

J	S_{j1}	S_{j2}	$\mu_{\tilde{U}}(J)$	$\mu_{\tilde{V}}(J)$
1	10	12	0.454545	0.545455
2	8	11	0.421053	0.578947
3	11	10	0.523810	0.476190
4	12	13	0.480000	0.520000

As we can observe the result of the algorithm is $\{2,1,4,3\}$. With assessing all the possible schedules the optimum makspan is equal to 32 when we have $\{2,1,4,3\}$ or $\{2,4,3,1\}$.

4-Performance analysis

To evaluate the performance of the proposed heuristic, we have used it on some problems in small sizes and the results are compared with optimum scheduling that result from assessing all the feasible solution. Notice that large size problems are NP hard. We have produced 49000 problems in 49 categories where they are formed by different quantities in the number of machines and number of jobs (table-4).

Table 4: Calculating $\left(C_{OPT} \div C_H\right)$

$\frac{C_{OPT}}{C_H}$		Number of machines						
		1	2	3	4	5	6	7
Number of jobs	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	2	1.0000	0.9971	0.9912	0.9884	0.9883	0.9877	0.9811
	3	1.0000	0.9956	0.9794	0.9772	0.9768	0.9727	0.9703
	4	1.0000	0.9955	0.9699	0.9592	0.9677	0.9575	0.9553
	5	1.0000	0.9951	0.9622	0.9525	0.9524	0.9462	0.9460
	6	1.0000	0.9950	0.9578	0.9425	0.9421	0.9413	0.9410
	7	1.0000	0.9949	0.9570	0.9411	0.9381	0.9351	0.9318

According to tabele-4, the proposed heuristic produces acceptable responses in comparison to the optimum solution. For more than 7 machine and 7 jobs calculations are very time-consuming.

5-Concluding remarks

For making more extensions of Johnson's rule for more than two machines we present a heuristic algorithm, using fuzzy sets. According to Johnson's rule all jobs are associated to one of two set $U=\left\{j \middle| t_{j1}<t_{j2}\right\}$ or set $V=\left\{j \middle| t_{j1}\geq t_{j2}\right\}$, in the other hand, $\mu_{\tilde{U}}(J)$ and $\mu_{\tilde{V}}(J)$ can only be 0 or 1 while in our fuzzy algorithm jobs membership is a fuzzy number between [0,1]. Proposed heuristic algorithms performance is analyzed in comparison to the optimum schedule in small size problems. Unfortunately application of the new algorithm could not be investigated for large size problems easily, because calculating the optimum solution is NP-hard.

6- References

1- S.M. Johnson (1954) Optimal two and three-stage production schedules with setup times included. Naval Research Logistics Quarterly 1: 69–81.

2- M.R.D. Garey, D.S. Johnson, R. Sethi (1976) The complexity of flow shop and job shop scheduling. Mathematics of Operations Research 1: 117–129.

3- A. Reisman, A. Kumar, J. Motwani (1997) Flowshop scheduling/sequencing research: a statistical review of the literature, 1952-1994. IEEE Transactions on Engineering Management 44:316-329.