BIO Web of Conferences **97**, 00146 (2024) https://doi.org/10.1051/bioconf/20249700146
ISCKU 2024

Some theorems of minimizing makespan and range of lateness of flow shop machines

Adel Hashem Nouri[1*], *Hussam Abid Ali* Mohammed[2], *and Kareema Abed* Al-Kadim[3]

[1]Department of Computer Science, College of Education, University of Kufa, Najaf, Iraq.
[2]Department of Mathematics, College of Education for Pure Sciences, University of Kerbala, Karbala, Iraq.
[3]Department of Mathematics, College of Education for Pure Sciences, University of Babylon, Hilla, Iraq.

Abstract: One of the most challenging and widely debated issues in scheduling theory is the topic of multi-objective job scheduling. The multi-objective function problem including the makespan and the range of lateness of three machines in a flow shop was explored in this study in order to solve it. Some of the theories discussed in this work have to do with the computation of the precise best time and range of lateness. Two types of the three machines have been created, and the theories for each type have been studied. These theories seek to cut down on the amount of time it takes to complete tasks and the range of lateness.

1 Introduction

One way to look at the scheduling problem is as a decision-making problem. It includes a combination of resources (machines), tasks (jobs), and objective criteria (objective criteria) that need to be combined under certain restrictions to create a mathematical formula with data. The assignment of machines, either one or many, to specific tasks over a period of time in a variety of manufacturing sectors periods is done on an organized basis with the intention of optimizing or minimizing (one or more) objective criteria.[1]

Machine Scheduling Problems (MSPs) are a part of the area of combinatorial optimization problems and are characterized as a method of making decisions that may be used often in a variety of service sectors and manufacturing. Its goal is to minimize one or more objectives. MSP deals with the allocation of resources to act across certain time periods. In a flow shop, n tasks must be completed on m machines in the same machine sequence. This implies that for all n tasks, the sequence in which they are handled on various computers is fixed and consistent. Each work might be processed on a different machine starting with machine 1, then machine 2, and so on until machine 3 is used to complete the final job.[2,3].

[*] Corresponding author : adilh.alhajjar@uokufa.edu.iq

2 Related works

Cheng and colleagues [4] investigated the two-machine flow-shop issue with release timings, where the goal is to decrease either the makespan or the maximum lateness. They construct strong new dominance orders that are embedded into branch-and-bound algorithms and provide a unified approach to many sequence interchange operators. In order to minimise makespan and decrease total completion time, Allahverdi [5] resolved the three machine flowshop scheduling issue by considering setup times independently of processing times, rather than assuming that they are part of processing durations.

The two-machine flowshop scheduling problem with batching and release time was researched by Tang and Liu [7]. The objective is to decrease the makespan. To resolve the scheduling issue and provide a bottom limit for the issue, they developed a mixed integer programming paradigm. Alternative procedures are possible to reduce the mean flow time and makespan of tasks, respectively, in a two-machine flow-shop scheduling problem studied by Lo and colleagues [8]. The aim function of the two-machine flow shop scheduling issue that Moslehi and colleagues [9] looked at is to minimize the total of the maximum earliness and tardiness. Some relevant lemmas were presented to build a branch-and-bound method. Pour and colleagues [10] offered a multi-objective approach to the flowshop problem, using the planner-specified weights as the objectives for makespan, total waiting time, and total tardiness.

Lin [11] took on the problem of making the most efficient use of processing times by minimising the makespan for a two-stage, three-machine assembly flow shop scheduling. an iterative greedy (IG) algorithm with four local search techniques and a cloud theory based CSA algorithm are included, along with a branch-and-bound approach, an LB, and several dominance qualities. A two-machine flowshop scheduling model was studied by Yang and Kuo [12]. In this model, the processing times of tasks are directly related to the durations that jobs must wait on the first machine before they can be processed on the second.

3 Important notations

n: Number of jobs.
p_{ij}: Processing time of job i on machine j.
d_i: Due date of jobs i.
s_i: Slack time of job i s.t. $s_i = d_i - p_{ij}$, where j is the last machine.
C_{ij}: Completion time of job i on machine j.
L_i: Lateness of $j$ob i, $L_i = C_{ij} - d_j$, where j is the last machine.
R_L: Range of lateness, $R_L = L_{max} - L_{min}$.

4 Machine scheduling problem

A few simple definitions will do for this study.

Definition (4.1) [13]: Suppose we have set of all schedules S for a scheduling problem FP, a schedule $s \in S$ is said is called **feasible** if it satisfies all the constraints of the problem FP.

Definition (4.2) [15] **(Shortest Processing Time (SPT) rule):** The order in which jobs are sequenced is determined by the processing time (p_i), which is expressed as (p_1≤p_2≤⋯≤p_n). The issue 1||∑C_i was solved using this rule.

Definition (4.3) [14] **(Earliest Due Date(EDD) rule):** In order to minimise the issue 1||Tmax, jobs are sequenced in a non-decreasing order of due date (d_i).

Definition (4.4) [15] **(Minimum Slack Time (MST) rule):** It is well-known that the issue 1||Emax may be solved using the rule of non-decreasing slack times (s_i), where s_i=d_i-p_i. This order is used to sequence jobs.

Definition (4.5) [15]: A solution where "optimise" is used in a problem involving multiple criteria for making a decision means that there is no way to improve one objective without making the other objective worse.

Definition (4.6) [13] (**Efficient Solution** (ES)): If no alternative schedule s' satisfies with at least one of the aforementioned conditions holding as a strict inequality $f_j(s') \leq f_j(s), j = 1, \dots, K$, then schedule s is considered efficient. Alternatively, s' is said to dominate s. Assume that σ is a schedule and that S is a collection of efficient schedules. If all efficient solutions of S do not dominate σ, we use the symbol $\sigma_{\vee}$S.

5 Problem formulation

For a sequence of jobs $s \in S$ where S is the set of all feasible solutions, the criteria (f_1) and (f_2) are computed in the following as:

Let $C_{max} = C_{n3}$ and $R_L = L_{max}(s) - L_{min}(s)$, $L_{max} = \max_{1\leq i\leq n}\{L_i\}$ and $L_{min} = \min_{1\leq i\leq n}\{L_i\}$ where $L_i = C_{i3} - d_i,\ i = 1, \dots, n.$

Our problem FP can formally by stated for any schedule $s \in S$ as:

$$\left.\begin{array}{ll} Min\begin{cases} f_1(s) = C_{max} & \dots (P1) \\ f_2(s) = R_L & \dots (P2) \end{cases} & \\ s.t. & \\ C_{i1} = \sum_{k=1}^{i} p_{k1} \dots\dots\dots\dots\dots\dots\dots\dots (1) & i = 1, \dots, n, \\ C_{12} = p_{11} + p_{12}, \dots\dots\dots\dots\dots\dots\dots\dots (2) & \\ C_{i2} = max\{C_{i1}, C_{i-1,2}\} + p_{i2}, \dots\dots\dots (3) & i = 1, \dots, n, \\ C_{13} = p_{11} + p_{12} + p_{13}, \quad \dots\dots\dots\dots\dots\dots (4) & \\ C_{i3} = max\{C_{i2}, C_{i-1,3}\} + p_{i3}, \dots\dots\dots (5) & i = 1, \dots, n, \\ L_i = C_{i3} - d_i, \quad \dots\dots\dots\dots (6) & i = 1, \dots, n. \end{array}\right\} (FP)$$

The issue being handled does not let the machine to be idle. We examine a multi-criteria issue where all performance indicators are minimised simultaneously. The hardness level of the flowshop problem F3‖ |(Cmax,R_L) is NP.

Some theorems and their corollaries are given for our issue, and two specific types of flow shops—Proportional Flow Shop (PFS) Type 1 and Type 2—are defined.

Assuming p_iA=p_i1+p_i2 and p_iB=p_i2+p_i3, within Type 1 $P\boldsymbol{OS}$, the processing time of any job on machine A is either less than or equal to the processing time of every job on machine B. Consider Type 1 PFS and its associated lemmas and corollaries.

Lemma (5.1): In Type 1 PFS, where the objective is to minimize C_{max}, the SPT_A-sequence gives OS for the objective C_{max} and the minimum value of the objective function C_{max} is:

$$p_{1A} = p_{11} + p_{12},$$

$$C_{max} = p_{1A} + \sum_{i=1}^{n} p_{i3}.$$

Proof: It is clear. ■

Corollary (5.2): In Type 1 PFS, if a job j has a minimum processing time on machine A, then must be scheduled first and any sequence with remaining jobs gives OS for the objective function C_{max}.

Proof: It is clear from Lemma 5.1. ■

Lemma (5.3): In Type 1 PFS, where the objective is to minimize R_L, for the two adjacent jobs i and j, if $d_i \le d_j$ and $d_i - p_{iB} \le d_j - p_{jB}$, then job i precedes job j in the optimal schedule.

Proof: Let $p_{iA} = p_{i1} + p_{i2}$ and $p_{iB} = p_{i2} + p_{i3}$, consider a schedule $s_1 = s'ijs''$ and a schedule $s_2 = s'jis''$ which is obtained by interchanging the jobs i and j in s_1, where s' and s'' are partial sequences of jobs and the two neighboring jobs i and j with $d_i \le d_j$ and $d_i - p_{iB} \le d_j - p_{jB}$. Let $t = p_{1A} + \sum_{k=1}^{i-1} p_{k3}$ the completion time for the last job of s'.

he proof of this lemma has to be shown that creating the sequence s_1 does not increase or decrease the values of L_max and L_min of the two jobs i and j, respectively, since the replacement of jobs i and j does not affect the completion times and, by extension, the lateness of other jobs. Based on the above relations, the following relations are proved:

(1) To prove $L_{max}(s_1) \le L_{max}(s_2)$ using the condition $d_i \le d_j$, let L is the maximum lateness for $(n-2)$ jobs of s' and s''.

For the schedule s_1, we have:

$L_{max}(s_1) = max\{L, L_i(s_1), L_j(s_1)\}$

$= max\{L, p_{1A} + \sum_{k=1}^{i-1} p_{k3} + p_{i3} - d_i, p_{1A} + \sum_{k=1}^{i-1} p_{k3} + p_{i3} + p_{j3} - d_j\}$

Now, consider the schedule s_2, we have:

$L_{max}(s_2) = max\{L, L_j(s_2), L_i(s_2)\}$

$= max\{L, p_{1A} + \sum_{k=1}^{j-1} p_{k3} + p_{j3} - d_j, p_{1A} + \sum_{k=1}^{j-1} p_{k3} + p_{j3} + p_{i3} - d_i\}$

Compare $L_i(s_2)$ with $L_i(s_1)$ and $L_j(s_1)$ where $d_i \le d_j$

It is clear that $L_i(s_2) > L_i(s_1)$ since $p_{j2} > 0$

Also, it is clear that $L_i(s_2) \ge L_j(s_1)$ since $d_i \le d_j$

Hence $L_i(s_2) \ge max\{L_i(s_1), L_j(s_1)\}$

That implies $L_{max}(s_1) \le L_{max}(s_2)$.

(2) To prove $L_{min}(s_1) \ge L_{min}(s_2)$ with condition $d_i - p_{iB} \le d_j - p_{jB}$, let L is the minimum lateness for $(n-2)$ jobs of s' and s''.

For the schedule s_1, we have:

$L_{min}(s_1) = min\{L, L_i(s_1), L_j(s_1)\}$

$= min\{L, p_{1A} + \sum_{k=1}^{i-1} p_{k3} + p_{i3} - d_i, p_{1A} + \sum_{k=1}^{i-1} p_{k3} + p_{i3} + p_{j3} - d_j\}$

Now, consider the schedule s_2, we have:

$L_{min}(s_2) = min\{L, L_j(s_2), L_i(s_2)\}$

$= min\{L, p_{1A} + \sum_{k=1}^{j-1} p_{k3} + p_{j3} - d_j, p_{1A} + \sum_{k=1}^{j-1} p_{k3} + p_{j3} + p_{i3} - d_i\}$

Compare $L_j(s_2)$ with $L_j(s_1)$ and $L_i(s_1)$ where $d_i - p_{iB} \le d_j - p_{jB}$

It is clear that $L_j(s_2) < L_j(s_1)$ since $p_{iB} > 0$

Also, it is clear that $L_j(s_2) \le L_i(s_1)$ since $d_i - p_{iB} \le d_j - p_{jB}$

Hence $L_j(s_2) \le min\{L_j(s_1), L_i(s_1)\}$

That implies $L_{min}(s_1) \ge L_{min}(s_2)$.

From (1) and (2) we have:

$L_{max}(s_1) - L_{min}(s_1) \le L_{max}(s_2) - L_{min}(s_2)$

Then $R_L(s_1) \le R_L(s_2)$.

Therefore the sequence s_1 is better than the sequence s_2 and job i precedes job j. ■

Corollary (5.4): In Type 1 PFS, if a job i with the minimum due date has the minimum slack time on machine B $(d_i - p_{iB})$, then the sequence with order EDD and MST_B minimizes the objective function R_L.

Proof: Directly from Lemma 5.2. ■

Example (5.5): If we have four jobs with the processing times $p_{i1} = (4, 3, 3, 2)$, $p_{i2} = (3, 4, 5, 1)$ and $p_{i3} = (7, 8, 6, 12)$ and the due date $d_i = (25, 24, 26, 27)$, the schedule $s = (1, 2, 3, 4)$ satisfy the conditions in Lemma 5.2 and Corollary 5.2. The schedule $s = (1, 2, 3, 4)$ with $R_L = 22$ is the smallest value in all FSs of S.

In Type 2 PFS, assume that $p_{iA} = p_{i1} + p_{i2}$ and $p_{iB} = p_{i2} + p_{i3}$, the processing time of any job on machine A is greator than or equal those of every jobs on machine B. The following lemmas and corollaries are presented for Type 2 PFS.

Lemma (5.6): In Type 2 PFS, where the objective is to minimize C_{max}, the LPT_B-sequence gives OS for the objective C_{max} and the minimum value of the objective function C_{max} is:

$$p_{nB} = p_{n2} + p_{n3},$$

$$C_{max} = p_{nB} + \sum_{i=1}^{n} p_{i1}.$$

Proof: It is clear. ■

Lemma(5.7): In Type 2 PFS, if a job j has a minimum processing time on machine B, then must be scheduled last and any sequence with remaining jobs give OSs for the objective function C_{max}.

Proof: It is clear from Lemma 5.6. ■

Theorem (5.8) [16]**:** There is an efficient sequence in the multi-criteria problem $P1$ that satisfies Johnsons-rule with condition either $min(p_{i1}) \geq max\,(p_{i2})$ or $min(p_{i3}) \geq max\,(p_{i2})$.

Corollary (5.9): In the OS for the objective C_{max}, if jobs $i \in s' \subset s \in S$ satisfy the Johnsons-rule condition with $min(p_{i3}) \geq max\,(p_{i2})$, then the subsequence s' sequencing by the SPT_A-sequence.

Proof: It is clear from Theorem 5.8. ■

Corollary (5.10): In the OS for the objective C_{max}, if jobs $i \in s' \subset s \in S$ satisfy the Johnsons-rule condition with $min(p_{i1}) \geq max\,(p_{i2})$, then the subsequence s' sequencing by the LPT_B-sequence.

Proof: It is clear from Theorem 5.1. ■

Theorem (5.11): If $d_j = d$, then the Johnsons rule is ES for $F3|d_j = d|(C_{max}, R_L)$ problem $P1$ with $(C_{max}, R_L) = (C_{max}(Johnsons), C_{n3} - C_{13})$.

6 Conclusion

The machine scheduling problem for multi-criteria objective function is one of the most difficult problems that have been discussed by many of researchers. In this paper, some theories are studied to solve the problem of multi-criteria objective functions consisting of makespan and the range of lateness of three machines in two types of proportional flow shop $F3||(C_{max}, RL)$, which is NP- hard. Some of these theories gave efficient solutions to our problem, and others gave us solutions to dominate some jobs over others.

References

1. M.L.Pinedo, "Scheduling Theory Algorithms, and Systems ", Third Edition.Stern School of Business, New York University, New York. Springer, (2008).
2. H.Holger. Hoos and S .Thomas, "In the Morgan Kaufmann Series in Artificial Intelligence", (2005).

3. I.k .Sun Lee, "A Scheduling Problem to Minimize Total Weighted Tardiness in the Two- Stage Assembly Flowshop", Mathematical Problems in Engineering, (2020), vol. 2020, Article ID 9723439, 10 pages.
4. Jinliang Cheng, George Steiner and Paul Stephenson, "Fast algorithms to minimize the makespan or maximum lateness in the two-machine flow shop with release times", JOURNAL OF SCHEDULING, (2002); 5:71–92 (DOI: 10.1002/jos.094).
5. Ali Allahverdi, "Three-Machine Flowshop Scheduling Problem to Minimize Total Completion Time with Bounded Setup and Processing Times",(2007), Decision Making in Manufacturing and Services, Vol. 1, , No. 1–2, pp. 5–23.
6. Ali Allahverdi, "Three-Machine Flowshop Scheduling Problem to Minimize Makespan WITH Bounded Setup and Processing Times", Journal of the Chinese Institute of Industrial Engineers, (2008). Vol. 25, No. 1, pp. 52-61.
7. Lixin Tang and Peng Liu, "Minimizing makespan in a two-machine flowshop scheduling with batching and release time", Mathematical and Computer Modelling, (March 2009Volume 49, Issues 5–6, , Pages 1071-1077.
8. G .Moslehi, M. Mirzaee, M.Vasei, M.Modarres, and A.Azaron, , "Two-Machine Flow Shop Scheduling To Minimize the Sum of Maximum Earliness and Tardiness",(2009), Int. J. Production Economics 122:763–773,.
9. Ming-Cheng Lo, Jen-Shiang Chen and Yong-Fu Chang, "Minimizing Makespan and Mean Flow Time in Two-Versatile-Machine Flow-Shop with Alternative Operations", (2010),Information Technology Journal, 9 (2): 257-265.
10. N. Shahsavari Pour , R. Tavakkoli-Moghaddam and H. Asadi, "Optimizing a multi-objectives flow shop scheduling problem by a novel genetic algorithm", International Journal of Industrial Engineering Computations, (2013), 4, 345–354.
11. Win-Chin Lin, "Minimizing the Makespan for a Two-Stage Three-Machine Assembly Flow Shop Problem with the Sum-of-Processing-Time Based Learning Effect", Discrete Dynamics in Nature and Society, (2018), Article ID 8170294, 15 pages https://doi.org/10.1155/2018/8170294.
12. Dar-Li Yang and Wen-Hung Kuo, "Minimizing Makespan in a Two-Machine Flowshop Problem with Processing Time Linearly Dependent on Job Waiting Time", Sustainability, (2019), 11(24), 6885; https://doi.org/10.3390/su11246885.
13. J.A Hoogeveen, "Single Machine Scheduling to Minimize a Function of Two or Three Maximum Cost Criteria, Journal of Algorithms, (1996), 21:415–433
14. L.Jouni, "Multi-Objective Nonlinear Pareto-Optimization" Lappeenranta University of Technology, (2000).
15. J.A .Hoogeveen, "Minimizing maximum earliness and maximum lateness on a single machine", Center for Mathematics and Computer science, (1991), P.O. Box 4079, 1009 AB Amsterdam, The Netherland,
16. Ibrahim M. Alharkan Algorithms for sequences and scheduling, Saudi Arabia, (2005).