E-ISSN: 2583-9667
Indexed Journal
Peer Reviewed Journal
https://multiresearchjournal.theviews.in

Received: 03-11-2023
Accepted: 13-12-2023

INTERNATIONAL JOURNAL OF ADVANCE RESEARCH IN MULTIDISCIPLINARY

Volume 2; Issue 1; 2024; Page No. 336-340

Permutation flow shop scheduling model

[1]NSV Kiran Kumar and [2]Dr. Manish KR Singh

[1]Research Scholar, Sunrise University, Alwar, Rajasthan, India
[2]Associate Professor, Sunrise University, Alwar, Rajasthan, India

DOI: https://doi.org/10.5281/zenodo.12664589

Corresponding Author: NSV Kiran Kumar

Abstract

Permutation flow shop scheduling is a significant problem in operations research and industrial applications, characterized by its complexity and relevance in optimizing production processes. This abstract provides an overview of the permutation flow shop scheduling model, its key components, challenges, and methodologies employed for solving it. The permutation flow shop scheduling model involves a series of machines (stages) through which jobs must pass in a specified sequence. Each job requires processing on each machine in a predetermined order, leading to various possible job sequences. The primary objective is to determine an optimal sequence that minimizes a specific criterion, such as makespan (total completion time) or total flow time. Key challenges in permutation flow shop scheduling include the combinatorial explosion of possible job sequences, non-preemptive processing constraints, and the need to balance workload across machines to achieve efficient utilization. Various heuristic, metaheuristic, and exact methods have been developed to tackle these challenges, including genetic algorithms, simulated annealing, and branch-and-bound algorithms. This abstract emphasizes the importance of permutation flow shop scheduling in enhancing production efficiency, reducing costs, and improving delivery times in manufacturing and service industries. It highlights ongoing research trends and future directions aimed at advancing solution methodologies, addressing real-world constraints, and integrating emerging technologies such as machine learning and Industry 4.0 concepts into scheduling optimization strategies.

Keywords: Scheduling, flow shop problem, optimization, characterized, permutation

Introduction

The flow shop scheduling model with the objective of optimizing the single criteria specifically makespan. Multistage ow shop scheduling problem with *m* stages each having distinct machines and with the goal of obtaining permutation schedule of *n* jobs corresponding to the performance criteria of minimum makespan can be presented as $F_m|permu|C_{max}$. Garey, Johnson and Sethi [GJS76] successfully reported that $F_m|permu|C_{max}$ is strongly NP-hard for $m \geq 3$ and therefore cannot be solved in the polynomial-time using exact methods. More- over, ow shop scheduling problems with the criteria of minimum makespan are combinatorial optimization problems that can be solved by approximate solution methods known as heuristics. The heuristics mentioned in ow shop scheduling literature can be categorized as constructive heuristics and improvement heuristics. Different authors have defined the concept of constructive heuristics in different ways. For instance, Lourenco [Lou96] defined constructive heuristics as the algorithms, solutions obtained, from which cannot be improved further, and in contrast improvement heuristics are the algorithms that begin with the initial feasible solution and can be improved iteratively to obtain the best approximate solution. However, Pinedo [Pin10] defined constructive heuristics as heuristics that serves as the basis to improvement heuristics.

For any production and manufacturing organization, the makespan is defined as the total time span between the start time of the first job and the completion time of the last job in any job schedule. Makespan has a direct influence on the utilization rate of production machines and equipment. Minimum elapsed time (makespan) ensures maximum resource utilization. Therefore, minimizing the makespan is the foremost objective of any industrial and manufacturing enterprise. The real-life significance of ow shop scheduling environment with criteria of minimum makespan is

successfully studied in the literature. Some of the tremendous studies, in this case, are analyzed as follows:
Erseven *et al*. reported the performance of permutation ow shop scheduling problem in the quality control department of a textile company. In the quality control department of a textile company, various test jobs arrive on different test stations of the production platform and considering these test stations as well as processing times of these jobs, a schedule has to be prepared. Each of these jobs has to ow through a series of operations that are performed by these test stations. These jobs have different processing times and have to wait on various stations to go through a series of quality control operations. The authors successfully developed a user-friendly and flexible tool that facilitates the company persons to prepare quick and efficient processing order of jobs that may result in the minimum value of makespan.
Kopanos, Mendez and Puigjaner reported that the pharmaceutical company manufactures a vast array of the products in a flexible ow shop environment consisting of four production stages granulation, compression, coating, and packaging, for each drug. These four stages reported for manufacturing may demand more than one workstation in the series.
Vob and Witt reported the study of sixteen production stages in the German steel manufacturing industry. According to the functionality of these steel industries, production jobs are created for every position in a customer order and for every stage in the ow shop environment. The main objective of the company is to generate new job schedules as often as required in order to complete all the events well in short time intervals within the production system and to meet the realistic due dates for new orders.

Materilas and Methods

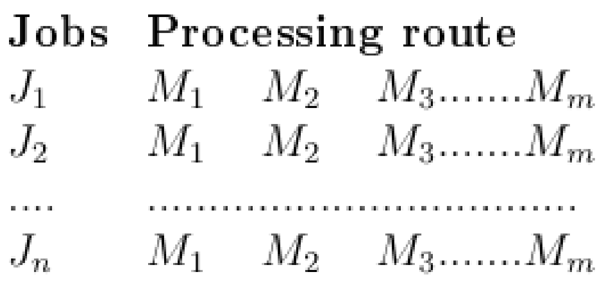
Ezzatollah Asgharizadeha, Paria Samadi Parviznejad (2021) [1] This study models the scheduling of an unbroken hybrid flow issue in the presence of uncertainty. We have used the fuzzy programming approach to regulate the parameters of processing time and preparation time in workshops, which helps with the unpredictability of processing time. A number of tasks in the suggested paradigm call both human and automated processing. The suggested model's primary goal is to minimise total completion time (Cmax) by identifying the optimal order of activities and allocating them to machines and workers in a stage-by-stage fashion. Additionally, this work utilises the fuzzy programming approach to regulate an unknown parameter in GAMS software in order to solve example issues. Findings from addressing problems in small and medium dimensions reveal that processing time and, by extension, completion time, rise in direct proportion to the level of uncertainty. Rises as a result of everything done. However, the time it takes to do all of the works has reduced as a result of the high efficiency of the machines, which has led to a rise in the number of machines and people at each stage. New features introduced in this study include a method for scheduling continuous hybrid flows of storage that takes into account not only payment time but also preparation and fuzzy processing time. One additional novel aspect of this piece is the way occupations are assigned to both humans and robots.

Kewal Krishan Nailwala, Deepak Gupta and Kawal Jeet (2016) [2] Jobs are continuously flowed through several equipment in a no-wait flow shop. Once begun, the work should be processed continuously via the machines without any waiting. This happens when workloads are processed sequentially on two different computers but no intermediate storage is available. Because NP-hardness is a difficulty in minimising makespan in flow shop scheduling, heuristic algorithms are essential for finding an ideal solution or a simple way to get closer to the optimal solution. In this work, we provide a heuristic approach for minimising makespan by modifying an existing heuristic, and a second heuristic algorithm for sequencing n-jobs through m-machines in a flow shop under a no-wait requirement. We compare the suggested heuristic algorithms to the NEH under no-wait and the MNEH heuristic for no-wait flow shop problem on 120 of Taillard's benchmark problems that have been published in the literature. By increasing the performance of NEH by 27.85%, MNEH by 22.56%, and the suggested constructive heuristic algorithm by 24.68%, the improvement heuristic surpasses all other heuristics on the Taillard's examples. The publication also includes numerical examples to clarify the algorithm's computing process. To arrive at these results, statistical tests of significance are conducted.
Janaki Elumalai *et al*. (2023) [3] At the operational decision-making level, job scheduling is a crucial responsibility of production logistics that helps organisations stay competitive. Using flow shop scheduling without task block criteria, processing times for multi machines are correlated with their probabilities. The end objective is to reduce the overall duration of all tasks. Finding the best or almost best sequence is Johnson's technique for reducing total elapsed time. It's easy to understand. The method is better understood with the assistance of a numerical demonstration.
Deepak Gupta, Sonia Goe (2018) [4] Scheduling n-jobs on three machines, with the first machine having m-parallel machines, is the topic of this work. You can see how much it costs to run each task on each parallel computer. Here you may get the processing time and probability of each task running on each computer. Finding the best possible task schedule that minimises the overall time it takes to complete each project is our goal.
Behnaz Zanjani, Amiri Maghsoud, Payam Hanafizadeh, Maziar Salahi (2021) [5] Allocating scarce resources to individual tasks in a manufacturing process is the goal of scheduling, a crucial decision-making process. In the realm of scheduling difficulties, Hybrid Flow Shop (HFS) scheduling is highly adaptable and has several real-world applications. This includes a multitude of scenarios where factors that impact task processing throughout execution are unclear. Accordingly, parameter uncertainty should be included by a good scheduling model. When dealing with real-world situations where processing time and due date are expected to be variable, this work introduces a multi-objective Robust Mixed-Integer Linear Programming (RMILP) model. Under unknown parameters, the created model may allocate tasks to available machines in a way that maximises the trade-off between two goals, such as total tardiness and makespan. To resolve this issue with several objectives, Fuzzy Goal Programming (FGP) is used.

Lastly, cases of varying sizes are produced and solved using the CPLEX solver of the GAMS programme under varying degrees of uncertainty in order to research and verify the effectiveness of the created RMILP model. In an HFS issue with processing time and due date uncertainty, the experimental findings demonstrate that the created model can identify a solution with the fewest adjustments.

Flow shop scheduling model

In a manufacturing setting with m machines, the ow shop scheduling model takes them all into account {M_1, M_2, M_3,.., M_m}, set up with a clear path and n tasks {J_1, J_2, J_3,..., J_n} so that these machines can process them. Determining the best sequence in which these tasks should be processed by the available machine system is the primary goal of the scheduler. A technical limitation of this setting is that all tasks must go in only one way across the system; for example, if job J_1 has to be handled by a machine M_i prior before the machine M_{i+1} ($\forall\ i = 1, 2,.., m - 1$) subsequently, this is also true of every other system task. In addition, whenever a work is finished on one computer, it must be added to the queue on the machine that follows it in the series. This limitation in technology may also be expressed as:



Jobs	Processing route		
J_1	M_1	M_2	M_3.......M_m
J_2	M_1	M_2	M_3.......M_m
....	....		
J_n	M_1	M_2	M_3.......M_m

According to the model just described, there are m.(n!). of potential work schedules. Technological limitations in the ow shop scheduling environment restrict the number of unique schedules to n!. Numerous researchers, including Garey, Johnson and Sethi [GJS76], Gonzalez and Sahni [GS78b], Pinedo [Pin10], and others, have recognised that even for medium-sized situations, this reduced number is still rather big and NP hard. The process shop scheduling issue for a system with m machines may be expressed as F_m/perm/γ when the scheduling triplet notation is used. Regarding γ eld, the usual goal functions, also known as performance metrics, can be thought of as reducing the overall time it takes to finish all the tasks (makespan), lowering the average tardiness (or lateness), minimising the maximum tardiness, optimising the quantity of jobs that are late, lowering the average ow time, and so on.

Proposed computational technique

The permutation ow shop scheduling problem with the objective of minimum makespan is NP-hard. Therefore, to obtain the best possible approximate solution of objective function following algorithm *NEHSMM* is proposed and implemented, by combining the tie-breaking strategy TB_{SMM} with NEH heuristic. The various steps are as follows:

Step 1 Compute the total processing time $T_j = \sum_{i=1}^{m} p_{ij}, \forall j = 1, 2, 3, ..., n.$

Step 2 Arrange all the jobs in the descending order of T_j and obtain the initial feasible job schedule (π).

Step 3 Consider the first two jobs from schedule π and arrange them corresponding to a minimum value of makespan to obtain the partial candidate-job schedule σ. In case of tie implement the proposed tie-breaking strategy TB_{SMM} however if a tie still exists, then the position of the jobs corresponding to first obtained partial schedule σ with a minimum value of makespan is considered. Set length of resultant job schedule as $L = 2$.

Step 4 Insert the next unscheduled job from initial feasible solution π in current candidate-job schedule σ and again arrange the jobs claiming the minimum value of makespan (elapsed time), without violating the order of already scheduled jobs. When the choice is to be made among the multiple partial job schedules with a minimum value of makespan, the proposed tie-breaking strategy (2.2.1) is implemented to determine the best partial job schedule. After insertion of new job increase the length L of job schedule σ by 1.

Step 5 Repeat step 4 until $L \geq n$.

Numerical Illustration

Consider the scheduling environment of 10 jobs and 5 machines with the predetermined processing time p_{ij} for each of job j on each of machine i presented in Table 1.

Table 1: Data of processing time of 10 jobs and 5 machines

Jobs→ Machines↓	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J10
M_1	79	40	48	16	38	76	73	34	38	32
M_2	67	40	93	23	90	13	85	6	6	11
M_3	10	57	49	19	57	99	40	27	35	11
M_4	48	21	11	2	73	98	20	53	28	34
M_5	52	54	79	38	3	55	85	21	44	27

Following the first step of the algorithm discussed, the total processing time for each job is $T_1 = 256$, $T_2 = 212$, $T_3 = 280$, $T_4 = 98$, $T_5 = 261$, $T_6 = 341$, $T_7 = 303$, $T_8 = 141$, $T_9 = 151$, $T_{10} = 115$.

Therefore the initial feasible job schedule is:

(J_6, J_7, J_3, J_5, J_1, J_2, J_9, J_8, J_{10}, J_4)

To determine the final job schedule σ follow step 2 to step 5. Consider the first two jobs J_6 and J_7 from the initial feasible job schedule. Arranging these jobs simultaneously two job schedules (J_6, J_7) and (J_7, J_6) are obtained with value of makespan 426 units and 450 units, respectively. Therefore, the partial candidate- job schedule σ = (J_6, J_7) with minimum value of makespan is considered for the next iteration.

Now pick the next job J_3 from the initial feasible job schedule and insert it in all the available positions of partial candidate-job schedule σ. The resultant job

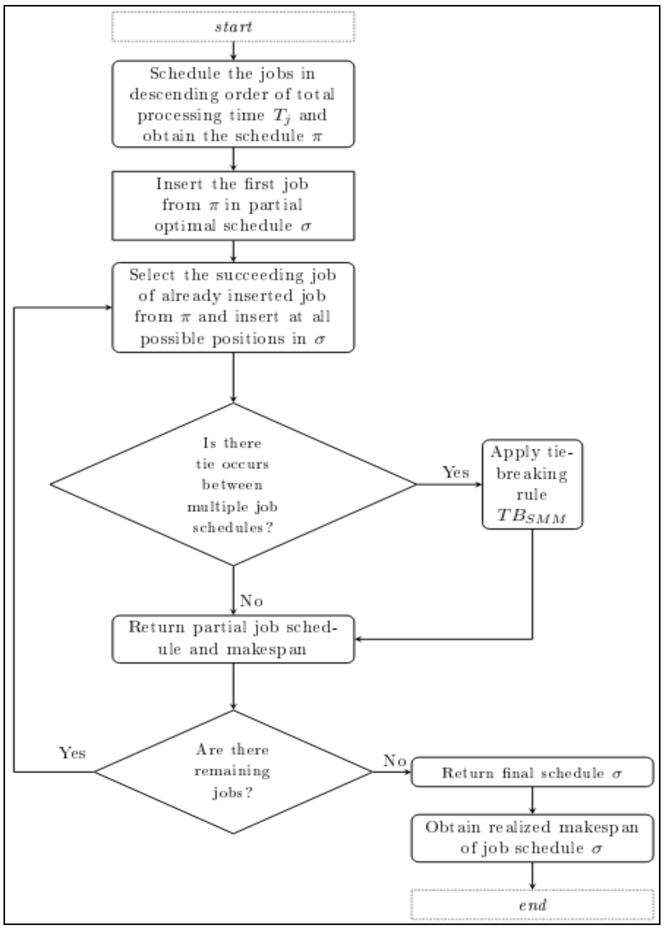

start
Schedule the jobs in descending order of total processing time T_j and obtain the schedule π
Insert the first job from π in partial optimal schedule σ
Select the succeeding job of already inserted job from π and insert at all possible positions in σ
Is there tie occurs between multiple job schedules?
Yes
Apply tie-breaking rule TB_{SMM}
No
Return partial job schedule and makespan
Are there remaining jobs?
Yes
No
Return final schedule σ
Obtain realized makespan of job schedule σ
end

Fig 1: Flow chart of the proposed heuristic

schedules are (J_3, J_6, J_7), (J_6, J_3, J_7) and (J_6, J_7, J_3). Here, the schedules (J_6, J_3, J_7) and (J_6, J_7, J_3) claims the minimum value of makespan, i.e. 505 units. Implementing the proposed tie-breaking rule the value of TB_{SMM} is obtained, 341.6 units and 358.4 units, for the job schedules (J_6, J_3, J_7) and (J_6, J_7, J_3), respectively. Therefore, the final schedule obtained by inserting job J_3, into the second position is marked as candidate-job schedule. As no extra loop is required to be implemented in the original code, therefore, computation of TB_{SMM} does not lead to an increase in complexity of the original NEH heuristic.

Implementing the various stages of the proposed heuristic the final job schedule is obtained as (J_4, J_2, J_{10}, J_6, J_3, J_1, J_7, J_8, J_9, J_5) with makespan 713 units which is comparatively less than the value 726 units obtained by the NEH heuristic.

Implementation of the tie-breaking mechanism in generating the final job sched- ule is presented in Table 2.

Table 2: The best partial schedules constructed by the improved NEH based heuristic after inserting each job of π

Iteration	Inserting job	σ	Cmax	Tie
1	J_7	(J_6, J_7)	426	no
2	J_3	(J_6, J_3, J_7)	505	yes
3	J_5	(J_6, J_3, J_7, J_5)	525	no
4	J_1	(J_6, J_3, J_1, J_7, J_5)	592	yes
5	J_2	(J_2, J_6, J_3, J_1, J_7, J_5)	632	yes
6	J_9	(J_2, J_6, J_3, J_1, J_7, J_9, J_5)	652	no
7	J_8	(J_2, J_6, J_3, J_1, J_7, J_8, J_9, J_5)	673	yes
8	$J10$	(J_2, J_{10}, J_6, J_3, J_1, J_7, J_8, J_9, J_5)	697	no
9	J_4	(J_4, J_2, J_{10}, J_6, J_3, J_1, J_7, J_8, J_9, J_5)	713	no

In-out ow of job schedule σ is presented in Table 3

Table 3: Flow table of jobs in schedule σ

Flow → Jobs ↓	M_1 In-Out	M_2 In-Out	M_3 In-Out	M_4 In-Out	M_5 In-Out
J_4	0-16	16-39	39-58	58-60	60-98
J_2	16-56	56-96	96-153	153-174	174-228
$J10$	56-88	96-107	153-164	174-208	228-255
J_6	88-164	164-177	177-276	274-374	374-429
J_3	164-212	212-305	305-354	374-385	429-508
J_1	212-291	305-372	372-382	385-433	508-560
J_7	291-364	372-457	457-497	497-517	560-645
J_8	364-398	457-463	497-524	524-577	645-666
J_9	398-436	463-469	524-559	577-605	666-710
J_5	436-474	474-564	564-621	621-694	710-713

Computational Experiment

To con rm the efficiency of the proposed tie-breaking rule and heuristic presented in solving the referred scheduling problem the computational experiment is carried out in MATLAB environment over Intel(R) processor *i*5 with 2.20 GHz, 8GB RAM on 64-bit Windows operating system. The constructive heuristics and metaheuristics specifically, DE [OD06], NEHKK1 [KK08], NEHKK2 [KK09], MOD [Mod12], CL$_{WTS}$ [YL13], GC [GC15], NEHAB1 [Bas16] are taken in reference for complete computational analysis in terms of two comparison tests including tie-breaking rule and performance of complete heuristics. The overall evaluation is carried out over standard Taillard's [Tai93] and VRF hard's [VRF15] scheduling benchmarks. These benchmarks comprise 600 problem instances with 60 combinations of jobs ($n \in \{10, 20, 30, 40, 50, 60, 100, 200, 300, 400, 500, 600, 700, 800\}$) and machines ($m \in \{5, 10, 15, 20, 40, 60\}$). The integer value of processing time for all these combinations is generated over the interval [1,99] following the uniform distribution.

The average relative percentage deviation (ARPD) is studied as the response variable in overall computational evaluation. Mathematically,

$$ARPD = \left(\frac{\sum_{k=1}^{10} (HS_k - BS_k)/BS_k}{10} \times 100 \right) \%$$

Here, HS_k represents the value of makespan obtained by the particular heuristic against the best approximate solution BS_k reported by Taillard [Tai93] and VRF [VRF15] for problem instance *k*.

Conclusion

In conclusion, a new heuristic NEHSMM is proposed and implemented to obtain the best approximate result of minimum makespan in permutation flow shop scheduling environment. The efficiency of the proposed heuristic has also been validated on both Taillard and VRF test beds. The future work on this strategy is to define more efficient techniques and tiebreaking strategies for the NEH heuristic so as to attain the best approximate results of makespan in the permutation flow shop scheduling environment.

Reference

1. Parviznejad PS, Asgharizadeh E. Modeling and Solving Flow Shop Scheduling Problem Considering Worker Resource. International Journal of Innovation in Engineering. 2021;1(4):1-7.
2. Nailwal K, Gupta D, Jeet K. Heuristics for no-wait flow shop scheduling problem. International Journal of Industrial Engineering Computations. 2016;7(4):671-680.
3. Elumalai J, Appasamy T, Ramalingam B, Joseph JP, Radhakrishnan K. Optimizing the completion time for flow shop scheduling model for multi machine and multi job without job block criteria. In AIP Conference Proceedings, AIP Publishing, 2023, 2790(1).
4. Gupta D, Goel S. Three stage flow shop scheduling model with m-equipotential machines. International Journal on Future Revolution in Computer Science & Communication Engineering. 2018;4(3):269-274.
5. Zanjani B, Maghsoud A, Hanafizadeh P, Salahi M. Robust multi-objective hybrid flow shop scheduling. Journal of applied research on industrial engineering. 2021;8(1):40-55.
6. Uysal F, İşleyen SK, Çetinkaya C. Resource constrained project scheduling with stochastic resources. Journal of applied research on industrial engineering. 2018;5(1):39-49.
7. Bougeret M, Pessoa AA, Poss M. Robust scheduling with budgeted uncertainty. Discrete applied mathematics. 2019;261:93-107. https://doi.org/10.1016/j.dam.2018.07.001
8. Rafiei A, Homayouni SM, Shafiei Alavijeh A. A Mathematical Model for the Single Machine Scheduling Considering Sequence Dependent Setup Costs and Idle Times. Journal of applied research on industrial engineering. 2015;2(2):77-85.
9. Ebrahimi M, Ghomi SF, Karimi B. Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. Applied mathematical modelling. 2014;38(9-10):2490-2504. https://doi.org/10.1016/j.apm.2013.10.061
10. Ribas I, Leisten R, Framiñan JM. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. Computers and operations research. 2010;37(8):1439-1454. https://doi.org/10.1016/j.cor.2009.11.001
11. Linn R, Zhang W. Hybrid flow shop scheduling: a survey. Computers and industrial engineering. 1999;37(1-2):57-61. https://doi.org/10.1016/S0360-8352(99)00023-6
12. Shiau DF, Cheng SC, Huang YM. Proportionate flexible flow shop scheduling via a hybrid constructive genetic algorithm. Expert systems with applications. 2008;34(2):1133-1143. https://doi.org/10.1016/j.eswa.2006.12.002
13. Wang H, Jacob V, Rolland E. Design of efficient hybrid neural networks for flexible flow shop scheduling. Expert systems. 2003;20(4):208-231. https://doi.org/10.1111/1468-0394.00245
14. Choi SW, Kim YD, Lee GC. Minimizing total tardiness of orders with reentrant lots in a hybrid flowshop. International journal of production research. 2005;43(11):2149-2167.
15. Choi SH, Wang K. Flexible flow shop scheduling with stochastic processing times: a decomposition-based approach. Computers and industrial engineering. 2012;63(2):362-373. https://doi.org/10.1016/j.cie.2012.04.001