Scientific Voyage
Vol. 1, No. 1
page 39-44 (2020)

SCIENTIFIC VOYAGE
ISSN: 2395-5546

ISSN: 2395-5546

A Discussion on Variants of Heuristic Algorithms for Flow Shop Scheduling Problem

Partha Haldar[1,*], Alok Mukherjee[2] and Kingshuk Chatterjee[3]

[1]*Mechanical Engineering, Government College of Engineering and Ceramic Technology, Kolkata 700 010, West Bengal, India*

[2] *Electrical Engineering, Government College of Engineering and Ceramic Technology, Kolkata 700 010, West Bengal, India*

[3]*Computer Science and Engineering, Government College of Engineering and Ceramic Technology, Kolkata 700 010, West Bengal, India*

Flowshop scheduling is a fundamental task in any industry related to production. An optimized schedule ensures efficient generation of products and minimization of waiting time. In this paper, we present a review of the different variants of the flowshop problem and also discuss some of the interesting heuristic algorithms employed to solve the practical problems.

Keywords: Flowshop problem, permutation, scheduling, continuous, block, heuristic algorithms.

1. Introduction

The flowshop scheduling problem involves n number of jobs and *m* no of machines. The jobs must pass through the m machines in a particular fixed order. The objective of the flowshop scheduling problem is to find a schedule which minimizes the total time of execution of the jobs (makespan). There are number of variations of the flowshop scheduling problem viz., permutation flowshop scheduling, continuous flowshop scheduling and block flowshop scheduling. The flowshop scheduling problem has been shown to be NP-hard which essentially means that an optimal solution to this problem is very difficult and is most likely to take exponential time. As a result, the algorithms that have been developed to solve the flowshop problem, try to obtain a near optimal solution using various methods such as heuristics, approximate algorithms and soft computing tools such as GAs. Many detailed survey of the flowshop scheduling problem are already present in the literature, and they are not up to date and do not discuss the latest algorithms associated with this field. In this paper, we incorporate the very recent heuristic based algorithms and works along with the already existing algorithms in this field. The main objectives of this paper are as follows:

1. To discuss the different variants of the flowshop scheduling problem
2. To discuss the different types of heuristic based algorithms employed to solve the flowshop scheduling problem.

2. Preliminaries

This section contains the basic definitions of the different variants of flowshop scheduling. These are briefed as follows:

2.1 Permutation flowshop scheduling:

In the case of permutation flowshop scheduling problem, there are n jobs that have to be performed on m unrelated
machines and every job consists of m non-preemptible operations. Every operation of a job uses a different machine

corresponding author: partha.haldar@gcect.ac.in

during a given time and may wait before being processed. For the permutation flow shop problem, the operations of every job must be processed on machines 1 , . . . , m in this order. Moreover, the processing order of the jobs on the
machines is the same for every machine. The problem consists in finding a permutation of the n jobs that minimizes the makespan.

2.2 Continuous flowshop scheduling:

The continuous flow-shop scheduling problem has the additional restriction that the processing of each job has to be continuous, i.e., once the processing of a job begins, there must not be any waiting times between the processing of any consecutive tasks of this job. Such a no-wait constraint is usually due to technological restrictions of the production process.

2.3 Blocked flowshop scheduling:

Blocked flowshop scheduling is a flowshop where a job having completed processing on a machine cannot leave this machine until its next machine is available for processing. That is, the job will block the current machine if its next machine is not available. The type of flowshop scheduling problem without intermediate buffers are known as the blocking flowshop scheduling problem. In the literature, the most common criterion for the blocking flowshop problem is the minimization of the makespan or the maximum completion time of the schedule.

3. Review of Heuristics Algorithms to solve flowshop scheduling problems:

In this section, we focus on the different heuristics techniques proposed by different authors to obtain near optimal solution to the flowshop scheduling problem. We discuss in brief, the different methods and modifications of some renowned heuristics algorithms in the next section. Heuristic Method is the most popular algorithmic technique to solve the flowshop scheduling problem. Various different types of heuristics have been used but the most popular heuristic that have been used is to arrange the jobs in ascending or descending order of their execution time.

Allahverdi et. al. [1] developed a new heuristics to minimize total completion time in m-machine flowshop problem. In this paper, the authors compare few heuristics that are independently developed and they propose several new heuristic methods modifying the existing heuristics models such as CDS heuristic [2], NEH heuristic [3], HO heuristic [4],Wang et al. heuristic [5], RZ heuristic [6] and WY heuristic [7]. Their analysis shows that a small proposed modification (pairwise exchange) improves the error performance of the best existing algorithms almost 50% with negligible CPU time.

Fink and Stefan [8] describes a method of solving the continuous flow-shop scheduling problem by metaheuristics. They examine the trade-off between running time and solution quality. They also investigate the knowledge and efforts needed to implement and calibrate their algorithms. They show that high quality results could be achieved in an efficient way by applying metaheuristics software components with neither the need to understand their inner working nor the necessity to manually tune parameters.

Framinan and Leisten [9] propose a heuristic for mean/total flowtime minimization in permutation flow shops. The heuristic exploits the idea of 'optimizing' partial schedules, already present in the NEH-heuristic [3] with respect to makespan minimization. Their experiment shows that their proposed heuristic model outperforms some existing best performing heuristics with respect to the quality of the solutions. The proposed model could further be embedded in an improvement scheme to build a composite heuristic for the flow time minimization problem.

Framinan and Leisten [9] represents an exhaustive review and comparative evaluation of heuristics and metaheuristics for the well-known permutation flowshop problem with the makespan criterion. They propose a comparison of 25 existing heuristics methods, ranging from the classical Johnson's algorithm or dispatching rules to the most recent metaheuristics, including taboo search, simulated annealing, genetic algorithms, iterated local search and hybrid techniques. The authors use the experimental design approach to validate the effectiveness and efficiency of the different methods.

Taillard [10] explains the problem of sequencing jobs in a permutation flow shop problem with an aim of minimizing the sum of flowtime, which is very much relevant to the latest dynamic production environment, and therefore it has attracted the attention of researchers during the last few years. Hence, a variety of heuristics have recently been developed, each claiming to be the best for the problem. The authors classify and conduct an extensive comparison among the existing heuristics, followed by analyzing the results of the experiments. They also suggest two new composite heuristics for the problem, which are proved very efficient for the problem.

Taillard [10] elaborates an efficient stochastic hybrid heuristic model for flowshop scheduling in order to minimize the makespan objective. Three probabilistic hybrid heuristics are proposed by the authors for solving the same and outperform best-known existing heuristics, combining elements from both constructive heuristic search and a stochastic improvement technique. Stochastic methods used by the authors include simulated annealing (SA). Tests are carried out to validate the improvements.

Liu and Reeves [11] explains another flowshop scheduling problem with the objective of minimizing the total flowtime using constructive heuristic and two composite heuristics methods. Computational analysis is carried out with the benchmark problems of Taillard [10]. The two proposed composite heuristic models prove better than the composite heuristics of Liu and Reeves [11]. Statistical tests of significance are used to validate the improvement in solution quality.

Chakravarthy et. al. [12] suggests a heuristic for scheduling in a flowshop with the objective of both minimizing the makespan and maximum tardiness of a job. The heuristic proposed by the authors makes use of the simulated annealing technique and compares the improvement of the proposal against the existing heuristic for scheduling to serve the objectives. The results of the computational evaluation reveal that the proposed heuristic performs better than the existing methods.

Laha et. al. [13] suggests a constructive heuristic for the permutation flowshop scheduling problem with an aim to minimize total completion time. Population-based technique has been adopted by the authors and also the insertion rule similar to that presented in the NEH [3] heuristic for the makespan criterion. The authors compare the computational results against the heuristics suggested by Woo and Yim [7] for small and large problem sizes and the heuristic of Framinan and Leisten [9] for small and medium problem sizes and suggests that the proposed heuristics performs better. The time complexity of the proposed method is also found to be less than those required by the existing topologies.

Pan et. al. [14] considers minimizing makespan for a blocking flowshop scheduling problem. A constructive heuristic is initially presented by the authors to generate a good solution by combining the existing profile fitting approach and NEH heuristic in an effective way. Further, the authors propose a Memetic Algorithm (MA) including effective techniques like a heuristic-based initialization, a path-relinking-based crossover operator, a referenced local search, and a procedure to control the diversity of the population, followed by calibration of the parameters of the proposed MA by means of a design of experiments approach. Finally, a comparative evaluation is conducted with the contemporary best performing algorithms for the blocking flowshop with makespan criterion, which justifies the effectiveness of the proposed MA algorithms over the other contemporaries.

A Distributed assembly permutation flow-shop scheduling problem is discussed by Lin and Shuai [15] which widely exists in modern supply chains and manufacturing systems. An effective hybrid biogeography-based optimization algorithm is proposed by the authors in this paper that integrates several novel heuristics with the objective of minimizing the makespan. Further, a novel local search method is designed based on the problem characteristics and included in the proposed algorithm scheme to further improve the outputs on 900 small-sized instances and 810 large-sized instances and validated to demonstrate the effectiveness of the proposed algorithm.

Benkalai et. al. [16] addresses the problem of scheduling a set of independent jobs with set-up times on a set of machines in a permutation flow shop environment. A metaheuristic known as the Migrating Birds Optimization is used by the authors for the minimization of the makespan. The authors present two versions of the algorithm, out of which, the first is a basic Migrating Birds Optimization and the second introduces some additional features. An extensive computational study is carried out to validate the efficiency of the proposed methods with some benchmark of instances. The authors found that the second version performs better and may be considered for solving real-world scheduling problems.

Deb et. al. [17] illustrates, a novel metaheuristic search algorithm namely rhinoceros search algorithm (RSA), which is inspired by rhinoceros' natural behaviour. The authors also relate the present proposal with their earlier proposal of another algorithm called elephant search algorithm. The authors show that RSA simplifies certain habitual characteristics of rhinoceros and stream-lines the search operations, thereby reducing the number of operational parameters required to configure the model as well as, is able to outperform certain classical metaheuristic algorithms. The RSA is also implemented on permutation flow-shop scheduling problem as well. Finally the authors also compare other optimization technique sand find some improvement of RSA over some of the existing topologies.

4. **Conclusion**

In this paper, we focus on the heuristic based flowshop scheduling which is justified because the optimized solution to the scheduling problem is supposed to take exponential time as it is an NP Hard problem. The primary focus of the survey is on the permutation flowshop variant. We have discussed various algorithms and their performances. In addition to the permutation flowshop problem, we have also discussed in details about the other flowshop variants such as continuous flowshop scheduling and block flowshop scheduling. Many researchers have had tried to develop algorithms to minimize the makespan, total flow time of the above stated variants of scheduling problems. Being NP Hard type problem, they need more and more effective and efficient solutions. The future researchers can try to use evolutionary algorithms, some nature inspired algorithms to solve this type of scheduling problem.

References:

1. A. Allahverdi and T. Aldowaisan, Int. J. Prod. Econ. **77** (2002) 71.
2. G.H. Campbell et al., Manage. Sci. **16** (1970) B-630.
3. M. Nawaz et al., Omega **11** (1983) 91.
4. C.J. Ho, Eur. J. Oper. Res. **81** (1995) 571.
5. C. Wang, C. Chengbin, and P. Jean-Marie, Eur. J. Oper. Res. **96** (1997) 636.
6. C. Rajendran and Z. Hans, Eur. J. Oper. Res. **103** (1997) 129.
7. H.S. Woo and Y. Dong-Soon, Comput. Oper. Res. **25** (1998) 175.
8. A. Fink and V. Stefan, Eur. J. Oper. Res. **151** (2003) 400.
9. J.M. Framinan and R. Leisten, Omega **31** (2003) 311.
10. E. Taillard, Eur. J. Oper. Res. **64** (1993) 278.
11. J. Liu, and C.R. Reeves, Eur. J. Oper. Res. **132** (2001) 439.
12. K. Chakravarthy and C. Rajendran, Prod. Plan. Cont. **10** (1999) 707.
13. D. Laha and A. Chakravorty, Int. J. Adv. Manuf. Tech. **53** (2011) 1189.
14. Q. Pan et al., IEEE Trans. Auto. Sci. Engg, **10** (2012) 741.
15. J. Lin, and Z. Shuai, Comput. Indus. Engg. **97** (2016) 128.
16. I. Benkalai et al., Int. J. Prod. Res. **55** (2017) 6145.
17. S. Deb et al., Soft Comput. **22** (2018) 6025