OPEN ACCESS
International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-7 Issue-5, June 2018

A Survey of The Optimization of The Flexible Job Shop Problem

Ghiath Al Aqel

Abstract: ***The flexible job shop problem is an important problem in modern manufacturing systems. It is known to be an NP-hard problem. The optimization of this problem can bring in considerable improvements in the manufacturing efficiency. In recent studies, it has attracted the attention of most researchers in this field. Several metaheuristic methods were proposed to solve this problem. These methods started with exact algorithms and later approximate methods, which include heuristic methods, evolutionary algorithms, swarm intelligence, local search and hybrid algorithms, were introduced to cope with the development and the growing scale of the flexible job shop problem. In this paper we explore the algorithms that are most commonly used to solve this problem. This paper also aims to evaluate and compare the performance of these algorithms.***

Index Terms: Flexible Job Shop, Optimization Algorithms.

I. INTRODUCTION

In modern competitive business environment, the companies aim to increase their profits by producing high quality products at low cost and high speed. Investing in assets, e.g. labor, tooling, machine, material and equipment, and employing these assets efficiently to satisfy the customer needs is the only way to stay in the market in long term at high profit levels. For the long term survival in the rapidly changing market, the manufacturing environments have to become more adaptive, or in other word flexible, supported by the multi-functional alternative machines to perform each operation. The flexible job shop scheduling problem (FJSP) is an extension of the job shop scheduling problem where operations are allowed to be processed on the multiple capable machines. The flexible job shop scheduling problem is formed by assigning each operation to a machine out of a set of capable machines and sequencing the assigned operations on each machine in order to obtain a feasible schedule by considering a predefined objective function.

The FJS problem assumes that each machine can process only one operation at a given time. Each operation is assigned to one of its capable machines and is processed on its assigned machine without interruption. There are precedence constraints that indicate the processing order between operations of a job. The operation cannot be performed before all its predecessor operations are finished. In this study, we also use ready time constraints and initial machine available time constraints. The ready time constraints exist when the jobs enter into production system at different times from the start of the scheduling horizon due to their arbitrary delivery times from the suppliers or subparts of the job continuing to be processed in other production departments. The initial machine available time constraints exist as the machines may become available at different times due to the maintenances required to start the batch of the scheduling horizon or due to the operations pending from the previous scheduling horizon. In the literature two types of the FJSP are mentioned according the speed of the capable machines. For type I FJSP, jobs can have alternative identical machines for each operation. The problem is to assign each operation to one of its identical capable machines and to find the sequence of operations on each machine. For type II FJSP, operations can have non-identical capable machines to be performed at. The problem is to assign each operation to one of its non-identical capable machines and arrange assigned operations at each machine.

Manuscript published on 30 June 2018.
***** Correspondence Author (s)
Ghiath Al Aqel*, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, E-mail: fisrtauthor@gamil.com

II. PROBLEM DEFINITION

For processing *n* jobs on *m* machines, the problem is to find the best solution that achieves the minimum or maximum value for an objective function. In the FJSP, there are a set of machines $A = M_1, \ldots, M_m$, and a set of jobs, $J = J_1, \ldots, J_n$ so that each job J_i consists of a given sequence of n_i operations, $O_{i,1}, O_{i,2}, \ldots, O_{i},n_i$. Each operation $O_{i,j}$, can be processed on any machine of a subset $A_{i,j} \subseteq A$ which represents the routing sub-problem. The other sub-problem is the sequencing sub-problem which is to sequence the operations on the machines. In this paper, the objective function is to minimize the makespan (maximal completion time) of all jobs.

In this research the following assumptions are considered:

1) All machines are available at time 0;
2) All jobs are released at time 0;
3) Each machine can process only one operation at a time;
4) Each operation can be processed without interruption on one of a set of available machines;
5) Recirculation occurs when a job could visit a machine more than once;
6) The order of operations for each job is predefined and cannot be modified.

III. LITERATURE REVIEW

Several methods have been used to deal with the FJSP. These approaches are classified into two major groups; the exact methods and the approximation methods [1].

Exact algorithms include mathematical programming (MP), while the approximation algorithms include some dispatching rules (DRs) and artificial intelligence (AI) based approaches. In the following, we briefly overview the most common used methods of both kinds to solve the FJS problem.

A. Exact Algorithms

[2] proposed a polynomial graphical algorithm when they first presented the FJSP with two jobs. [3] evaluated some mathematical models of the FJSP.

Due to the complexity of the problem, the exact methods require much longer computational time to obtain the optimal solution. The CPU time of the obtained solution increases exponentially with the increase of the problem size. Hence, exact methods are usually unsuitable for real life problems. However, [4] has also proved that exact algorithms are ineffective when dealing with large scale problems of FJSP. Consequently, these research directions have slowly fallen falls into elimination, despite of the significant contribution made in early research work.

B. Approximate Methods

Approximate methods have evolved constantly by time, forming a significant alternative approach. Studies have proven the efficacy and the accuracy of the approximate methods. Their development has a long and rich history, and varies from dispatching priority rules through shifting bottleneck approach, geometric approach, job insertions, local neighborhood search, neural networks, and most recently simulated annealing, tabu search, evolutionary algorithms and swarm intelligence. The significant development of these methods has allowed to efficiently solve instances within an operation size of less than 2000 [5].

1) Heuristic Methods

In the FJS problem, as part of scheduling problem, the CPU time that is needed to obtain an approximate solution (a near optimal solution) is much less than the CPU time needed to obtain an optimal one. This fact expresses the main advantage of using heuristic methods over exact methods. In addition, the near optimal solution could be adequate for many of the real-life FJS problems. Some recent studies focused on developing heuristic algorithms that merely obtain near-optimal solutions in the shortest possible time. The most frequently studied heuristics, dispatching rules (DRs), are simple decision-making procedures for an operation to select a machine of a possible set of machines that is capable of processing that operation. In literature, DRs have been studied extensively in solving the FJS problem. [6] applied some existing DRs to solve the routing sub-problem and used several different TS methods to deal with the sequencing sub-problem. It's been indicated by [7] that an integrated approach of the DRs might provide better results, although it is still more difficult to be adopted in real systems. [8] analyzed the effects of several DRs on the scheduling performance of job shops with different levels of flexibility and with different sizes of the problem. They proved that the performances of these DRs were approximately similar when dealing with high machine flexibility. On the other hand, different performances were obtained for zero machine flexibility. [9] created an auto-selection of combined DR by converting them into measurable contribution factors in the optimizing process of scheduling problems. [10] developed a dynamic DR to solve the machine scheduling problem. They used a new priority measure, and proposed an effective heuristic. [11] analyzed the DR evolved via genetic programming, and suggested not to use a wide range of number of jobs and machines for the instances in the training set. [12] compared different ensemble learning methods for creating DR and proposed ensembles of DRs that was able to outperform the ones obtained by Genetic programming. [13] Adopted GA to explore the best combination of DR and used a "Hold" strategy for multi-objective JSS.

2) Evolutionary algorithms (EA)

Evolutionary algorithms are an effective type of meta-heuristic methods. For instance, the Genetic Algorithm (GA), which mimics the natural process of evolution, is one of the most frequently used algorithms in the optimization of scheduling problems. The process analyzes the fitness of a species in their environment. Along different generations, it aims to maintain the good characteristics and eliminates the poor ones. Simultaneously, GA uses diverse strategies to decide which part of an individual to keep, and which one to discard. Such methods consist of several functions which we usually call "genetic operators". Genetic operators most commonly include; mutation, selection, and crossover. The selection operator is set to distinguish between good and bad solutions in order to produce more solutions from good solutions with promising fitness values. The crossover operator involves two individuals from the selection step as the parents for generating new solutions. Finally, mutation operator is responsible of varying or expanding the search process and by exploring new neighborhoods in order to avoid being trap in a local optimal. GA usually dictates initial population. In literature, the initial population is either generated randomly, or created by some initializing heuristics. Some of these heuristics analyze the characteristics of the instance and hence they are more likely to result in better solutions. The literature the FJS problem is rich of GA approaches. [14] proposed a method for learning and evolving of FJSP named LEarnable Genetic Architecture (LEGA). [15] Proposed a bi-level GA in an attempt to keep the advantages of preceding generations and reduce the disturbance of genetic operators. Later, an improved GA were proposed by [16] targeting a better initialization and faster convergence. [4] proposed a GA which included diverse strategies for initializing the population and individuals' selection for reproducing. [17] solved the FJSP with operation's overlapping and used an effective GA. [18] proposed a multi-stage operation-based GA to solve the problem from of a dynamic-programming perspective. [19] proposed a modified GA that was effectively able to deal with distributed scheduling problems and the FJSP. [20] developed an improved GA using opposition-based learning. The method used a multi-parent precedence operation crossover and a modified neighbor search mutation with opposite inverse mutation [21].

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Also presented an improved GA by using a knowledge-based system taken from the flexible job shop production system, in which, operations sequence is fixed for each job and each operation can be assigned to a selected machine of a work station.

A new method was developed by [22] based on a hierarchical multi-space competitive GA. This optimization approach separates complex problems into a set of simpler problems, where each sub-problem is optimized independently. [23] focused on initializing the population for GA and proposed a new method.

3) Swarm Intelligence (SA) based algorithms

Swarm intelligence (SI) algorithms mainly include ant colony optimization (ACO), particle swarm optimization (PSO) algorithm, and artificial bee colony (ABC). [24] presented a system based on ACO for the FJSP with routing flexibility and sequence-dependent setup and transportation time. [25] focused on reducing the total weighted tardiness in JSSP and combined the artificial bee colony (ABC) algorithm with neighborhood search. They designed a tree search algorithm to enhance the utilization abilities of the ABC. This new technique was considered to be quite effective in solving the job shop scheduling problem with the total weighted tardiness benchmarks.

[26] presented an artificial bee colony algorithm (ABC) that reflected the balance between global and local search. They used combined strategies to create initial solutions with a guaranteed quality and variety. Then, the operators of crossover and mutation are employed to create the neighboring food for the bees for both routing and sequencing sub-problems. Finally, a local search method that is based on critical path is added. Their experiments showed satisfactory performance when compared with methods proposed by [27], [28], [29], and [30].

4) Local Search methods (LS)

Local search methods try to obtain an optimum within a local area of the entire solutions space of an FJS problem, this optimum is normally a near-optimal solution. The quality of the obtained optimum strongly depends on the initial solution. Moreover, the design of the neighborhood structure contributes directly to the efficiency of the method [1]. Some heuristics are often used for the initialization process. The searching process begins with initialization, and then searching around the current solutions, which is done by making some changes to this solution. The main drawback of these methods is that they tend to get trapped in a local optimum of the search neighborhood. This, in some cases, could result in a solution quality far from the global optimum.

5) Tabu Search (TS)

The main feature that separates tabu search from other local search methods is its adaptive memory. The TS method depends on a memory that preserves the tabu moves, which are the recently obtained solutions. This helps to avoid the trap of local optimum and attains the balance between exploitation (reinforcing the search in neighboring areas) and exploration (exploring new regions of the solution space). The fitness value of every move in each iteration is assessed, and the position with the best quality is accepted and stored as the current solution. The mentioned procedure faces some challenges. For instance, the increment of the size of the tabu list leads to consuming longer CPU time, while reducing that size might not lead the convergence towards further improvement of the solution quality. Therefore, the size of the tabu list must achieve this balance to get a satisfactory solution quality in an acceptable CPU time.

6) Hybrid algorithms

Many researchers attempted to combine several algorithms to create some effective hybrid algorithms for FJSP. Many of the presented hybrid algorithm have provided satisfactory results in that they gain the advantages of more than one single method by coupling these methods together.

Reference [31] combined GA with TS and used a time-varying crossover and time-varying maximum step size of search in order to control the convergence of the local search into an optimal solution. They tested the proposed algorithm on many benchmark instances and compared the results with those obtained by [14]. The comparison indicates that their algorithm outperformed the latter in all instances.

Reference [30] presented a novel TS method hybridized with a public critical block neighborhood structure. The used three different insert/swap functions to form three different approaches in the neighborhood structure for the routing sub-problem. These approaches helped in the reduction of the neighborhood size by skipping infeasible moves. The results proved that their algorithm is competitive in both solution quality and convergence speed.

IV. CONCLUSION

In this paper optimization algorithms for solving the FJS problem have been explored. GA is the most commonly used algorithm for solving the FJS problem. The application of swarm intelligence (including ACO, BCO, and ABC) for the FJS problem growing in recent years while the use of exact algorithms is being much weakened. The use of hybrid algorithms is also growing recently especially the hybridizing of GA with other methods.

REFERENCES

1. Li, X. and L. Gao, An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. International Journal of Production Economics, 2016. **174**: p. 93-110.
2. Brucker, P. and R. Schlie, Job-shop scheduling with multi-purpose machines. Computing, 1990. **45**(4): p. 369-375.
3. Demir, Y. and S.K. İşleyen, Evaluation of mathematical models for flexible job-shop scheduling problems. Applied Mathematical Modelling, 2013. **37**(3): p. 977-988.
4. Pezzella, F., G. Morganti, and G. Ciaschetti, A genetic algorithm for the flexible job-shop scheduling problem. Computers & Operations Research, 2008. **35**(10): p. 3202-3212.
5. Taillard, E., Benchmarks for basic scheduling problems. european journal of operational research, 1993. **64**(2): p. 278-285.
6. Paulli, J., A hierarchical approach for the FMS scheduling problem. European Journal of Operational Research, 1995. **86**(1): p. 32-42.
7. Zhang, G., L. Gao, and Y. Shi, An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Systems with Applications, 2011. **38**(4): p. 3563-3573.
8. Baykasoğlu, A. and L. Özbakır, Analyzing the effect of dispatching rules on the scheduling performance through grammar based flexible scheduling system. International Journal of Production Economics, 2010. **124**(2): p. 369-381.
9. Ingimundardottir, H. and T.P. Runarsson, Evolutionary learning of linear composite dispatching rules for scheduling, in Computational Intelligence. 2016, Springer. p. 49-62.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

10. Ying, K.-C., S.-W. Lin, and C.-C. Lu, Effective dynamic dispatching rule and constructive heuristic for solving single-machine scheduling problems with a common due window. International Journal of Production Research, 2017. **55**(6): p. 1707-1719.
11. Mei, Y. and M. Zhang. A comprehensive analysis on reusability of GP-evolved job shop dispatching rules. in Evolutionary Computation (CEC), 2016 IEEE Congress on. 2016. IEEE.
12. Đurasević, M. and D. Jakobović, Comparison of ensemble learning methods for creating ensembles of dispatching rules for the unrelated machines environment. Genetic Programming and Evolvable Machines, 2017: p. 1-40.
13. Huang, J. and G.A. Süer, A dispatching rule-based genetic algorithm for multi-objective job shop scheduling using fuzzy satisfaction levels. Computers & Industrial Engineering, 2015. **86**: p. 29-42.
14. Ho, N.B., J.C. Tay, and E.M.-K. Lai, An effective architecture for learning and evolving flexible job-shop schedules. European Journal of Operational Research, 2007. **179**(2): p. 316-333.
15. Peigen, Z.C.R.Y.L. and S. Xinyu, BILEVEL GENETIC ALGORITHM FOR THE FLEXIBLE JOB-SHOP SCHEDULING PROBLEM [J]. Chinese Journal of Mechanical Engineering, 2007. **4**: p. 020.
16. Peigen, Z.G.G.L.L. and Z. Chaoyong, Improved Genetic Algorithm for the Flexible Job-shop Scheduling Problem [J]. Journal of Mechanical Engineering, 2009. **7**: p. 026.
17. Demir, Y. and S.K. İşleyen, An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations. International Journal of Production Research, 2014. **52**(13): p. 3905-3921.
18. Zhang, H.P. and M. Gen, Multistage-based genetic algorithm for flexible job-shop scheduling problem. Journal of Complexity International, 2005. **48**: p. 409-425.
19. Jia, H., et al., A modified genetic algorithm for distributed scheduling problems. Journal of Intelligent Manufacturing, 2003. **14**(3): p. 351-362.
20. Huang, M., W. Mingxu, and L. Xu. An improved genetic algorithm using opposition-based learning for flexible job-shop scheduling problem. in Cloud Computing and Internet of Things (CCIOT), 2016 2nd International Conference on. 2016. IEEE.
21. Purnomo, M.R.A., A Knowledge-Based Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem. International Business Management, 2016. **10**(19): p. 4708-4712.
22. Ishikawa, S., R. Kubota, and K. Horio, Effective hierarchical optimization by a hierarchical multi-space competitive genetic algorithm for the flexible job-shop scheduling problem. Expert systems with applications, 2015. **42**(24): p. 9434-9440.
23. Jorapur, V.S., et al., A promising initial population based genetic algorithm for job shop scheduling problem. Journal of Software Engineering and Applications, 2016. **9**(05): p. 208.
24. Rossi, A. and G. Dini, Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method. Robotics and Computer-Integrated Manufacturing, 2007. **23**(5): p. 503-516.
25. Zhang, R., S. Song, and C. Wu, A hybrid artificial bee colony algorithm for the job shop scheduling problem. International Journal of Production Economics, 2013. **141**(1): p. 167-178.
26. Wang, L., et al., An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. The International Journal of Advanced Manufacturing Technology, 2012. **60**(1): p. 303-315.
27. Kacem, I., S. Hammadi, and P. Borne, Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2002. **32**(1): p. 1-13.
28. Zhang, G., et al., An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. Computers & Industrial Engineering, 2009. **56**(4): p. 1309-1318.
29. Xing, L.-N., et al., A knowledge-based ant colony optimization for flexible job shop scheduling problems. Applied Soft Computing, 2010. **10**(3): p. 888-896.
30. Li, J.-Q., et al., A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. The international journal of advanced manufacturing technology, 2011. **52**(5-8): p. 683-697.
31. Zhang, G., Y. Shi, and L. Gao. A genetic algorithm and tabu search for solving flexible job shop schedules. in Computational Intelligence and Design, 2008. ISCID'08. International Symposium on. 2008. IEEE.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)