Indian Journal of Science and Technology, *Vol 9(41), DOI: 10.17485/ijst/2016/v9i41/103955, November 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Scheduling on a Flexible Job Shop with Setup Operation in IT Manufacturing Fabrication

Sang-Oh Shim*

Department of Business Administration, Hanbat National University, Daejeon 34158, South Korea; soshim@hanbat.ac.kr

Abstract

Objectives: We consider the scheduling problem in which the separate sequence dependent setup is considered even though a job to be processed is not arrived. **Methods/Statistical Analysis**: In a flexible job shop considered here, to minimize average flow times and tardiness, heuristic algorithms based on dispatching rules are suggested. A series of computational experiment is done for evaluating the performance of the suggested ones and result shows that they perform better than modified methods in previous researches and a method used in a real situation. **Findings**: Priority rule based scheduling algorithms and relaxed mixed integer programming for obtaining the initial schedules and improvement procedures are suggested and they give very good solution in a reasonable time so that it can be argued that they are innovative and suitable for the sustainability on the manufacturing firms. This study can be extended to various researches. For example, different types of measurements, such as make span can be used in this problem. Also, one may extend to the large sized problems, i.e., larger number of jobs, operations and machines, for reflecting more real situation. On the other hand, one might need to develop the optimal methodologies such as combinatorial optimization ones. **Improvements/ Applications**: The suggested methodologies here can be used in IT manufacturing fabrication such as semiconductor and liquid crystal display ones to enhance the ability of production and customer satisfaction.

Keywords: Flexible Job Shop, Heuristics, IT Manufacturing, Scheduling, Sequence Dependent Setup

1. Introduction

A flexible job-shop reflecting real situation of IT manufacturing firms such as semiconductor and liquid crystal display is extended from a traditional job shop. In the flexible job shop, there are jobs, which include sequential tasks/operations, and machines. Each task/operation should meet precedence constraints between them. In this problem, schedules of each operation of job are obtained, that is, the assignment and sequence of each task/operation on each machine would be determined by considering separate setup times (sequence dependent), ready times and due dates of jobs.

In case of scheduling problem on flexible job-shop, few researches exist. In[1] devise polynomial time method for 2-job problem and heuristics by using it. Also[2], uses priority rules for assigning jobs to the machine and meta-heuristic for sequencing jobs. On the other hand, author in[3,4] propose a genetic algorithm and greedy search algorithm for the problem considered here, respectively. Meta heuristics are mostly used for solving the flexible job shop scheduling problem. For example, a tabu search algorithm is used by[5]. Also, author in[6] presents a genetic algorithm based on a dynamic programming.

There are lots of studies for sequence-dependent setup. For example, author in[7,8] suggest a mixed integer program for job shop scheduling problem with separate sequence dependent setup. On the other hand, author in[9] devise a genetic algorithm for the multi-criteria objectives.

Since the issues of scheduling jobs are raised in the current IT manufacturing firms, it is very important to schedule jobs effectively and efficiently[10–12]. Therefore,

**Author for correspondence*

in this flexible job-shop scheduling problem with property of separate sequence dependent setup, a scheduling algorithm based on the priority rules is suggested to minimize average flow time and average tardiness of jobs. Remaining part of the research is shown as following. In section 2, problem description is given and the suggested algorithm is shown in Section 3. With randomly generated test problems reflecting real situations, a series of tests is conducted in Section 4. In the final chapter, this paper is concluded with a short summary.

2. Problem Description

The motivation of this research is from the one of the processes at display manufacturing fabrications, especially color filter process. There are *m* parallel machines which can perform all the operations and *n* jobs which consist *k* operations. In this problem, those operations are ordered to complete job, that is, all operations can be processed in predetermined order for each job. In the color filter process of the liquid crystal display manufacturing fabrications, five operations, usually named as red, green, blue, lens and pads process, should be done on any photolithography machines, in the fixed order. Since the production cost and lead time of the color filter process usually are very high, it is very important to maximize production quantity and minimize the lead time by developing efficient and effective scheduling of jobs.

A setup operation with sequence dependent is required between the different operations and frequent set up is not preferred since it increases the lead time due to its relatively longer time than processing time of the operation. Especially, in this paper, separate setup time is considered, that is, setup operation can be processed if a machine becomes available even though a job to be processed is not arrived. Hence, if we know the next operation to be processed, set up can be started although there is no corresponding job. Also, these jobs are dynamically arrived, that is, they have different ready times, and they have different due dates.

Under the situation described above, the objective of this paper is developing scheduling jobs and their operations on each parallel-machine to minimize average flow times and average tardiness of jobs. Additional assumptions for solving the problem considered here, are 1. At time zero, all machines are idle, 2. No preemption is allowed and, 3. Only one operation can be done on a machine at a time. As performance measures, average flow time and average tardiness are computed as sum of difference between arrival and completion times of jobs and sum of difference between due dates and completion times of jobs, respectively. Since average flow time is concerned with work-in-process inventory and the average tardiness is related with customer satisfaction in the practical field, these two measures are used in this study. To compute performance measure, sum of α•average flow time and (1−α)•average tardiness is used. Here, α is set to 0.5 by preliminary tests.

3. Proposed Algorithms

In the suggested algorithms, an initial schedule is obtained based on the five priority rules, that is, when a machine becomes idle, an operation of a job with the highest priority value is selected and assigned to the machine. In addition to priority rules, a relaxed mathematical programming is developed to obtain the initial schedule. With the initial schedule, an improvement procedure is done.

Usually, for the different objectives, different priority rules are used among lots of ones. Hence in this study, to minimize average flow time and average tardiness, well known priority rules for them are used to obtain initial schedule. To use the priority rules notations used in this research are given first as following.

i index of job, $i = 1, 2, \ldots, n$.

j index of machine, $j = 1, 2, \ldots, m$.

d_i due date of job i.

s_t known sequence-dependent setup time at t-th operation.

t scheduled time, i.e. time when machine become idle.

a_i arrival time of job i.

o_{it} index of t-th operation of job i, $t = 1, 2, \ldots, k$.

p_{it} processing time of operation o_{it}.

r_{it} remaining processing time of job i including t-th operation which can be computed as $\sum_{t}^{k} p_{it}$.

θ_t binary variable, 1, if setup is required for t-th operation at the current partial schedule, otherwise 0.

S_t total setup time of remaining operations including t-th operation of all jobs which can be computed as $\sum_{t}^{k} s_t \cdot \theta_t$.

The first priority rule is Modified Due Date with Set up times (MDDS) rule[13] which is computed as max (d_i, $t + s_t + r_{it}$). Therefore, at each time to schedule, that is, when a machine becomes idle, for all waiting jobs to be processed on the machine, each priority for each operation associated with the job are calculated and selected. The second one is Modified Operation due Date with Set up time (MODS) rule, that is also developed by[13]. It is obtained by computing max {d_i - b•(r_{it}- p_{it}), $t + s_t + r_{it}$}, where b is empirically determined parameter, called the lead time estimator. The third rule is Cost Over Time (COVERT) rule[14], which can be computed as max [0, {1- max(0, d_i– t–r_{it}) / (k_1•b•r_{it})}] /p_{it}, where k_1 is a look-ahead parameter which also is predetermined like b. The fourth rule, Modified Apparent Tardiness Cost with Set up (MATCS) rules, in which ATCS developed by[15] is modified, is used and it is calculated as exp {– max(0, d_i – t – r_{it}) / (k_1• $\bar{p}$) – s_t / (k_2• $\bar{s}$)}/p_{it}, where k_2 is scaling parameter which can be determined empirically and $\bar{p}$ and $\bar{s}$ are mean processing and set up time of operations, respectively. ATCS rule[15] is originally from ATC rule proposed by[16] and is used in single machine scheduling problem with setup property. In addition to the four existing priority rules described above, one is proposed to consider multi-objectives, i.e., average flow times and tardiness. The fifth rule, suggested here and named as Apparent Tardiness and Remaining Setup times (ATRS) rule, is computed as exp {– max (0, d_i – t – r_{it}) / (k_3• $\bar{p}$) • S_t / (k_4• $\bar{s}$)}/p_{it} for operation o_{it}, where k_3 and k_4 are also scaling parameters determined empirically. Since two performance measures are considered here, two terms are used in the proposed priority rule. The first term, max (0, d_i – t – r_{it}) implies that slack of the job to the due date, hence it is expected that average tardiness can be improved. The second terms, St, shows the expected setup time suppose that operation o_{it} is scheduled at the current time. Therefore, it can reduce the average flow time. By using these two terms, the new priority rule is developed. All of five priority rules are computed for all operations which would be processed on the idle machine, and then the highest priority operation is chosen and assigned. Also, scheduling procedure would be repeated until all operations are scheduled. Then five initial schedules by using each priority rule can be obtained.

Besides the initial schedules results from the five priority rules described above, a relaxed Mixed Integer Programming (MIP) is used. Since the original problem, MIP, can be only solved in short time for small-sized problem, it is relaxed in a way that the objective of relaxed MIP is to minimize average flow times not considering average tardiness and it only considers that groups of same operations, in which has no setup requirement, are considered as decision variables to be scheduled. To solve the MIP, additional notations are used as follows.

S_{jq} start time of q-th ordered group on machine j
C_{jq} completion time of q-th ordered group on machine j
s_{it} start time of t-th operation of job i
X_{ijtq} decision variable, 1 if t-th operation of job i is processed in q-th ordered group on machine j, otherwise 0

Y_{tjq} decision variable, 1 if t-th operations are in q-th ordered group on machine j, otherwise 0

$$\text{Minimize} \quad Z = \sum_{i=1}^{n} \frac{s_{ik} + p_{ik} - a_i}{n} \qquad (1)$$

$$\text{Subject to} \quad s_{i1} \geq a_i \quad i = 1, 2, \ldots, n \qquad (2)$$

$$s_{it} \geq s_t + \sum_j \sum_q X_{ijtq} \cdot S_{jq}$$
$$i = 1, 2, \ldots, n,\ t = 1, 2, \ldots, k \qquad (3)$$

$$s_{it+1} \geq s_{it} + p_{it} \quad i = 1, 2, \ldots, n,\ t = 1, 2, \ldots, k \qquad (4)$$

$$S_{jq+1} \geq C_{jq}$$
$$j = 1, 2, \ldots, m,\ q = 1, 2, \ldots, n \bullet k \qquad (5)$$

$$C_{jq} \geq X_{ijtq}(s_{it} + p_{it}) \quad i = 1, 2, \ldots, n,$$
$$j = 1, 2, \ldots, m,\ t = 1, 2, \ldots, k,\ q = 1, 2, \ldots, n \bullet k \qquad (6)$$

$$C_{jq} \geq S_{jq} + s_t + \sum_i X_{ijtq} \cdot p_{it} \quad j = 1, 2, \ldots, m,$$
$$t = 1, 2, \ldots, k,\ q = 1, 2, \ldots, n \bullet k \qquad (7)$$

$$Y_{tjq} \geq X_{ijtq} \quad i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m,\ t = 1, 2, \ldots, k,\ q = 1, 2, \ldots, n \bullet k \qquad (8)$$

$$\sum_j \sum_q X_{ijtq} = 1 \quad i = 1, 2, \ldots, n,\ t = 1, 2, \ldots, k \qquad (9)$$

$$\sum_t Y_{tjq} = 1 \quad j = 1, 2, \ldots, m,\ q = 1, 2, \ldots, n \bullet k \qquad (10)$$

Minimizing average flow time, the objective of relaxed MIP, is shown in Constraint (1). Constraint (2) and (3) mean that the first operation of each job should be started after its arrival and that the start time of *t*-th operation should be greater than sum of its setup time and start time of *q*-th ordered group, respectively. Constraint (4) implies that the next operation should be started after end of its previous operation. The completion time of q-th ordered group should be less than the start time of the next ordered group in constraint (5). Constraint (6) and (7) ensure that the completion time of q-th ordered group on machine j should be greater than sum of the start time of t-th operation and its processing time or sum of start time of q-th ordered group, setup time and processing time. Finally, in Constraint (8), (9) and (10), assignments of job and its operations in ordered group on each machine are developed. In this paper, by solving the mathematical programming described above, the sixth initial schedule is obtained. To solve it, ILOG/CPLEX 10.0 is used. Here, the initial scheduled obtained by this mathematical model is denoted as RMIP.

After obtaining the initial schedules by using five priority rules and mathematical model, the improvement procedure is performed in a way that it does not deteriorate number of setups in the initial schedule, considering sequence of operations on each machine. From initial schedule, consecutive operations without setup requirement on each machine are re-sequenced in ascending order of their slack times while keeping the precedence constraints. This improvement step is repeated until there is no improvement of performance measure on each machine. Notice that average flow time of jobs is rarely affected by this re-sequencing due to considering only consecutive operations without setup requirement while average tardiness can be reduced. And then for the two operations which is required setup operation, their positions are exchanged on each machine if there is improvement of the performance measure. Note that the feasibility of schedule should be kept.

4. Computational Experiments

To figure out performance of suggested algorithms, 900 test problem instances (50 problems for each 18 combinations) are generated randomly, 18 combinations were made from three levels for the number of jobs (30, 60 and 90), three levels for the number of machines (3, 6 and 9), and two levels for the number of operations (5 and 10). From the discrete uniform distribution, each processing time of a job is generated between 10 and 60, and the setup time is set to the processing time multiplied by the number, which is generated by using discrete uniform distribution with range [1, 3] since it is relatively longer than that of processing time in real situation. The due date of jobs is set to the number in which sum of arrival time, total operation's processing time and mean setup time (total setup time divided by the number of machines) is multiplied by the randomly generated number between 1 and 5. This random number is due date tightness parameter. Also, the parameter, μ, in the exponential distribution for the inter arrival time between consecutive jobs is set to 10. Through the preliminary tests, parameters for priority rules, b, k_1, k_2, k_3 and k_4 are set to 1, respectively. All methodologies presented here were programmed in C++ language, and computational tests were conducted on a PC with an i3-4030U CPU operating at 1.9 GHz.

To see performance of the suggested algorithms, Percentage Reduction (PR) is calculated as $100\bullet(P_e - P_a)/P_e$, where P_e is the best solution value among the ones obtained by existing methods, four priority rules (MDDS, MODS, COVERT and MATCS) described above without the improvement procedure, and P_a is the solution value of algorithm *a* which are the suggested methods with the improvement procedure, i.e., the solution value obtained by the improvement procedure applied to each initial schedule after obtaining initial one by all priority rules, MDDS, MODS, COVERT, MATCS and ATRS, and mathematical model, RMIP, respectively. The larger percentage reduction, the better performance of the suggested algorithm. Also, the number of the best solution is used as performance measure. For each combination of test data, the number of best solution is counted. Hence, the larger number of best solution of the algorithm, it implies that the algorithm shows better performance.

The overall results are given in Table 1. All the suggested algorithm, i.e., MDDS, MODS, COVERT, MATCS, ATRS and RMIP with the improvement procedure, perform better than the best method among, MDDS, MODS, COVERT and MATCS without the improvement, that are well known in the real manufacturing system since the all average percentage reductions are greater than zero. Most of all, the newly suggested priority rule, ATRS, shows the best performance since it shows larger percentage reduction and number of best solutions than others. Also, it seems that the suggested improvement procedure works well since the percentage reduction is increased when the improvement procedures are conducted for the best schedule obtained by MDDS, MODS, COVERT and MATCS. In addition to solution quality, each algorithm with priority rule requires less than 0.1 seconds to solve each problem instance although the computation times are not given in Table 1. Unfortunately, although RMIP, the mathematical model shows better performance than other algorithms based on the priority rules except for ATRS, it takes too much time to obtain the solution. For the small sized problem of 30 jobs computation time is around 10 minutes. On the other hand, for the large sized problem of 90 jobs, it takes 60 minutes. However, except for the computation time of RMIP, this study proposes the good priority rule and improvement method that can be used in real flexible job shop to minimize average flow time and tardiness. Regardless of the various sizes of jobs, machines and operations, since the suggested algorithms give the better performance consistently, it can be argued that they can be used for various situations in real manufacturing systems since they have robustness property.

Even if the ATRS (Apparent Tardiness and Remaining Setup times) rule, suggested here shows the best performance, MATCS and COVERT rules show better results than MDDS and MODS rules except RMIP due to much computation time. They show similar performances and MODS rule shows better than MDDS rule. Hence, it can be summarized that 1. The best algorithm is ATRS rule with the improvement procedure, 2. The next algorithms are MATCS and COVERT rules with the improvement procedure, 3. The fourth is MODS rule with the improvement procedure and 4. The final one is MDDS rule with the improvement procedure. In this study, the t-test results between algorithms are not shown, since it is shown that there are significant differences between ATRS and MATCS (COVERT), MATCS (COVERT) and MODS, MODS and MDDS, respectively in Table 1.

5. Conclusion

Scheduling problem on flexible job-shop with separate sequence-dependent setup is addressed to minimize average flow time and tardiness in this study. Priority rule based scheduling algorithms and relaxed mixed integer programming to obtain the initial schedules and improvement procedures are suggested and they give very good

Table 1. Overall results of the suggested algorithms

Number of jobs	Number of machines	Number of operations	MDDS		MODS		COVERT		MATCS		ATRS		RMIP	
30	3	5	9.1†	3††	11.6	9	15.6	13	15.7	14	21.1	27	18.4	20
		10	5.4	0	10.8	6	12.4	10	11.6	11	20.9	26	17.5	18
	6	5	8.4	6	9.5	11	11.0	13	12.9	11	23.2	28	20.6	16
		10	6.7	8	12.4	15	13.1	18	15.4	16	26.5	29	23.1	19
	9	5	8.9	1	10.8	6	12.5	10	12.4	11	23.5	24	22.4	13
		10	8.5	2	9.6	5	11.6	8	10.9	10	22.4	26	18.9	16
60	3	5	9.5	6	10.4	10	12.8	16	13.6	18	24.7	24	15.8	15
		10	8.4	4	9.0	6	16.7	11	17.4	10	20.6	26	18.0	20
	6	5	3.1	0	8.4	7	14.6	11	13.9	13	23.4	28	16.9	21
		10	4.9	3	10.0	9	11.5	16	12.4	15	26.7	30	15.7	23
	9	5	5.9	6	9.5	10	13.9	14	14.6	13	22.1	24	16.4	18
		10	6.8	4	8.6	9	14.8	13	14.1	15	21.9	26	17.6	17
90	3	5	7.5	2	12.6	8	15.9	16	15.8	14	25.4	21	20.9	16
		10	6.0	0	8.5	4	11.5	9	12.6	13	23.7	26	16.7	20
	6	5	9.8	1	12.6	12	14.6	17	13.4	18	21.6	22	20.9	18
		10	10.1	6	13.9	11	15.3	13	16.8	15	26.7	24	24.1	16
	9	5	6.5	5	9.5	8	10.9	15	12.1	16	25.9	21	18.4	17
		10	8.7	6	13.2	9	16.8	14	17.6	16	26.0	20	22.0	16

† Average percentage reduction for 50 problem instances.

†† number of best solution obtained by the algorithm.

solution in a reasonable time so that it can be argued that they are innovative and suitable for the sustainability on the IT manufacturing firms.

This study can be extended to various researches. For example, different types of measurements, such as makespan can be used in this problem. Also, one may extend to the large sized problems, i.e., larger number of jobs, operations and machines, for reflecting more real situation. On the other hand, one might need to develop the optimal methodologies such as combinatorial optimization ones.

6. Acknowledgement

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2010-327-B00205).

7. References

1. Brucker P, Schlie R. Job-shop scheduling with multi-purpose machines. Computing. 1990; 45(4):369–75.
2. Brandimarte P. Routing and scheduling in a flexible job shop by tabu search. Annals of Operations Research. 1993; 41(3):157–83.
3. Kacem I, Hammadi, S, Borne, P. Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems. IEEE Transaction Systems, Man, and Cybernetics. 2002; 32(1):1–13.
4. Mati Y, Rezg N, Xie X. An integrated greedy heuristic for a flexible job shop scheduling problem. IEEE International Conference on Systems, Man and Cybernetics. 2001; 4:2534–9.
5. Hurink J, Jurish B, Thole M. Tabu search for the job shop scheduling problem with multi-purpose machines. Operations Research Spectrum. 1994; 15(4):205–15.
6. Yang JB. GA-based discrete dynamic programming approach for scheduling in FMS environments. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2001; 31(5):824–35.
7. Choi IC, Choi DS. A local search algorithm for job-shop scheduling problems with alternatives operations and sequence-dependent setups. Computers and Industrial Engineering. 2002; 42(1):43–58.
8. Zhou Y, Li B, Yang J. Study on job shop scheduling with sequence-dependent setup times using biological immune algorithm. International Journal of Advanced Manufacturing Technology. 2006; 30(1):105–11.
9. Guimaraes KF, Fernandes MA. An approach for flexible job-shop scheduling with separable sequence-dependent setup time. IEEE International Conference on Systems, Man and Cybernetics. 2006; 5:3727–31.
10. Srikanth GU, Maheswari VU, Shanthi AP, Siromoney A. Task scheduling model. Indian Journal of Science and Technology. 2015; 8(7):33–42.
11. Ashwin KSV, Rahul R, Dheepan P, Sendhil KKS. An optimal ant colony algorithm for efficient VM placement. Indian Journal of Science and Technology. 2015; 8(2):156–9.
12. Rishita K. Application of multi-core parallel programming to a combination of ant colony optimization and genetic algorithm. Indian Journal of Science and Technology. 2015; 8(2):138–42.
13. Baker KR, Bertrand JWM. A dynamic priority rule for scheduling against due-dates. Journal of Operations Management. 1982; 3(1):37–42.
14. Caroll CD. Heuristic sequencing of jobs with single and multiple components [PhD thesis]. Cambridge, MA: MIT; 1965.
15. Lee YH, Bhaskaran K, Pinedo M. A heuristic to minimize the total weighted tardiness with sequence-dependent setups. IIE Transactions. 1997; 29(1):45–52.
16. Vepsalainen APJ, Morton TE. Priority rules for job shops with weighted tardiness costs. Management Science. 1987; 33(8):1035–47.