

An Exact Method to Solve Scheduling and Production Smoothing Problem in the Flow Shop Environment

Mehdi Bijari , Parvin Mostofian

**Department of Industrial Engineering
Isfahan University of Technology, 84156-83111, Isfahan, Iran**

Abstract

Minimizing the makespan in the Flow shop scheduling problem with sequence dependent set-up times is considered in this paper. Production smoothing, as the important condition in JIT production systems, is also considered as a constraint in the problem. The problem is known to be NP-hard; therefore a Branch and Bound algorithm and a heuristic algorithm are proposed for solving the model. The proposed heuristic algorithm has been utilized for solving large scale problems. Computational results by 2040 test problems demonstrate that the efficiency of proposed algorithms. The proposed B&B algorithm is able to reach to optimum solution the problems with maximum 14 positions in a few minutes.

Keywords

Flow shop Scheduling Problem, Minimizing Makespan, Production Smoothing, Sequence Dependent Set-up Times

1. Introduction

Flow shop scheduling problem has a lot of applications in industry. Many researchers have worked and are working on this subject so a lot of papers are being published in this field in scientific journals every year. Most of these papers are about minimizing makespan in flow shop production environment. But, more and less, researches with other objectives or with specific hypothesizes are done in such production environments [1]. Production smoothing as an important role in JIT production environments has attracted the interests of many researchers [2]. However, a few researches have been done to consider production smoothing in flow shop environments. McMullen [3] introduces the concept of production smoothing for manufacturing systems where the final stage of manufacturing operations is a flow shop with sequence dependent set-up times.

This research presents a model to the permutation flow shop problem where Just in Time (JIT) production requirements are taken into account. McMullen developed a bi-criteria model. In addition to the traditional objective of minimizing the production makespan, minimization of Miltenburg's material usage rate is also incorporated. In this model, multiple units of each product are permitted in the production sequence. However, the minimization of material usage rates attempts to prevent batch scheduling of products and allows unit flow of products as required in demand flow. A solution method is proposed for determining an optimal production sequence via an efficient frontier approach and Simulated Annealing (SA). Test problems and specific performance criteria are used to assess the solutions generated by the proposed method. Experimental results presented in this paper show that the proposed method obtain optimal solution.

Yavuz et al. [4] is concerned with a batching problem encountered in the context of production smoothing in just-in-time manufacturing systems. The manufacturing system of interest is a multi-level system with a flow-shop at the final level. A hybrid meta-heuristic method is developed to solve the batching problem, which is known to be NP-hard. Strategic Oscillation (SO) and Path Re-linking (PR) methods have hybridized and compared the hybrid method's performance to two benchmark methods: a bounded dynamic programming method developed for the problem earlier and an implementation of Robust Tabu Search (RTS) meta-heuristic. Through a computational study, it is shown that the proposed hybrid method is effective in solving the problem within several minutes of computer time and yielding near-optimal results.

Yavuz et al. [5] is concerned with a special class of mixed-model manufacturing systems: flow shops. The production smoothing problem is studied under presence of non-zero setup times. The master problem is split into two sub-problems which are concerned with determining the batch sizes and production sequences, respectively. A dynamic programming procedure is developed to solve the batching problem, and using an existing method from the current literature is suggested to solve the sequencing problem.

Kogan and Tell [6] focus on advance orders and continuous-time production smoothing under uncertain demands. Yalcinkaya & Durmusoglu [7] consider lean based production system. In this study, production smoothing decisions are studied through value stream mappings. Smoothing scenarios for seasonal variation are

developed in order to reduce the deterioration of the production systems performance. Firstly the work flow was visualized using value stream mapping, then production schedule was created using production smoothing method and lastly developed smoothing scenarios were tested.

The above mentioned information shows there are a few researches on flow shop scheduling problem by considering production smoothing. In the other way, McMullen [3] was searching for a single flow of products. This research avoids batch production. McMullen [3] has presented a heuristic method based on SA algorithm to solve the problem and it has not searched for finding the optimal solving or best heuristics and this is an opportunity for future researchers.

2. Problem Definition

In this paper, the same as McMullen[3], scheduling problem of n jobs on m machines in a flow shop production environment is considered. The jobs have sequence dependent set-up times on the machines. Each job i has demand d_i . Production positions for sequencing of jobs are equal to $D = \sum d_i$. In this way, each job occupies one position in sequence. This helps to access to a smooth production sequence. Only one job can process on each machine, per time. Job interruption is not allowed.

Below notation is used to modeling the problem:

M: set of machines $\{k = 1, \dots, m\}$

J: set of jobs $\{i = 1 \dots n\}$

d_i : demand of job i, $D = \sum_{i=1}^n d_i$

P: set of positions in sequence $\{j = 1, \dots, D\}$

t_{ik} : process time of job i on machine k

$s_{ii'}$: set-up time of job i if it's done after job i'

z_{ij} : decision variable; it equals to 1 if job i occupies position j, else it equals to 0

α : index of sequence production smoothing

CSF_i : smoothing constant of job i, $CSF_i = d_i/D$

C_{jk} : completion time of job of position j on machine k

The problem model is:

$$\text{Min } C_{Dm} \quad (1)$$

$$\sum_{i=1}^n z_{ij} = 1 \quad \forall j \quad (2)$$

$$\sum_{j=1}^D z_{ij} = d_i \quad \forall i \quad (3)$$

$$c_{jk} = \max\{c_{j,k-1}, c_{j-1,k}\} + \sum_{i=1}^n t_{ik} z_{ij} + \sum_{i=1}^n \sum_{i'=1}^n s_{ii'} z_{ij} z_{i'j-1} \quad \forall j, k \quad (4)$$

$$\left[\sum_{i=1}^n \sum_{j=1}^D \left(\frac{\sum_{r=1}^j z_{ir}}{\sum_{r=1}^j \sum_{i=1}^n z_{ir}} - CSF_i \right)^2 \right]^{\frac{1}{2}} \leq \alpha \quad (5)$$

$$z_{ij} = 0 \text{ or } 1, t_{ik} > 0, s_{ii'} \geq 0 \quad (6)$$

Equation (1) is objective function of makespan minimization (completion time of the last job on the last machine). Equation (2) guaranties that only one job assign to each position. Equation (3) guaranties that each job occupies positions exactly equal to number of its demand. Equation (4) shows the computation phrase for completion time in each position. Equation (5) guaranties that summation of deviation of smoothing constant for all jobs is less than a determined amount of α . α is an index for determining smoothing level of deferent production sequences. This parameter is determined by the manager with due attention to existing requirements and conditions in real production environment. This modeling shows that the problem is NP-hard. Because a few increase in n and D parameters contributes to a large increasing in problem dimensions. This clears that is not efficient to use complete enumeration for solving this problem and it's necessary to find more suitable solving methods.

3. Solving Methods

A useful method for solving such problems is shortened enumeration strategy of solving called B&B algorithm. Now, usage of B&B algorithm for solving the problem is being described. Suppose σ is a detail sequence of jobs made by arranging some jobs of n jobs. Also, σ_i is a detail sequence that job i is immediately after sub-sequence σ and the last job in σ is i' . If P_{σ}^j shows a detail problem in the level j of branching tree, this will be the same problem P^0 that the j first positions of the sequence is assigned. $C_{\sigma(k)}$ and b_{σ} of P_{σ}^j show the completion time on machine k th and the portion of deviation of smoothing index for the detail sequence. That is:

$$C_{\sigma(k)} = \max\{C_{\sigma(k)}, C_{\sigma_i(k-1)}\} + t_{ik} + s_{ii'}$$

$$b_{\sigma_i} = b_{\sigma} + \sum_{l=1}^n \left(\frac{\sum_{r=1}^{j+1} z_{lr}}{j+1} - CSF_l \right)^2$$

In each stage of branching, possible branches are equal to the number of n jobs that are not assigned all their demands. Each detailed problem $P_{\sigma_i}^{j+1}$, made by choosing a job as i from the rest of the jobs.

In bounding process a lower bound for completion time is being searched. Here, $C_{\sigma(m)}$ is the mentioned lower bound. Now, it should be clear that completion of the detail problem P_{σ}^j can contribute to an optimal solving or not. In per time, b_{σ} is being computed and compared with α (smoothing index). If $b_{\sigma} > \alpha$, then completion of σ won't be desirable.

Here, backtracking method for branching has been used in order to occupy less memory space in computer and to simplify coding the problem.

α is a parameter being computed with attention due to production environment requirements by the management. For computational experiments, it should be computed, to apply in B&B algorithm. Here, a heuristic approach is used to compute deferent amount of α in test problems. It's clear that α is being determined with attention due to existing conditions in the real environments and it's not necessary to use this approach.

As mentioned before, the problem is NP-hard. So, a heuristic algorithm can be used to solve the problem in suitable time. Here, a heuristic algorithm is predicted to access to rather good solving, to compare with B&B algorithm results. This algorithm is based on the well-known nearest neighborhood approach in solving the traveling salesman problem.

Heuristic algorithm steps

Step 0: $j = 1$, $C = 0$ and $b = 0$;

Step 1: all jobs are being placed in list 1 (non-sequenced);

Step 2: C_{ij} and b_{ij} are being computed for all jobs in list 1;

Step 3: the job with smallest C_{ij} is being placed in position j , if two or more C_{ij} have been equaled then the job with smallest b_{ij} is being selected, in case of re-equation, the job is being selected randomly;

Step 4: $C = C + C_{ij}$ and $b = b + b_{ij}$ (i is the index of the selected job at step 3);

Step 5: $j = j + 1$ and the assigned job is being omitted from list 1;

Step 6: the 2nd to 5th steps are being repeated till when the list 1 will be empty.

4. Results

Some test problems by a computer program have been designed and solved to consider the performance of proposing algorithms. Process times have uniform distribution in [1,100]. Set-up times have uniform distribution in [1,10]. Number of Machines are considered 3, 10, 50 and 100. Number of jobs are natural numbers equal to and greater than 3 and number of sequence positions (the summation of all demands of all jobs) are 6, 8, 9, ..., 14, 15. Each specific problem with determining m , n and D is being generated 10 times with random process and set-up times. Parameter α is being computed in three ways for applying in test problems. The B&B algorithm is being performed with each amount of α on each test problem. Then the algorithm is being performed without α parameter to compare the results. Maximum time for performing the algorithm has limited to 5 minutes. Then each problem is being solved by heuristic algorithm.

All computations have done in a notebook with 1.7 GHz processor, 512 MB of ram memory. The coding has done in Visual Basic under Windows XP. Due to limited time of algorithm, increasing the number of m , n and D are being continued to number that can be solved in 5 minutes. The solved problems are:

Group 1: $m=3$ $\cdot n=3$ $\cdot D = \{6,8,9,10,11,12,13,14\}$; Group 2: $m=3$ $\cdot n=4$ $\cdot D = \{8,9,10,11,12,13,14,15\}$;

Group 3: $m=3$ $\cdot n=5$ $\cdot D = \{10,11,12,13,14,15\}$; Group 4: $m=10$ $\cdot n=3$ $\cdot D = \{6,8,9,10,11,12,13,14\}$;

Group 5: $m=10$ $\cdot n=4$ $\cdot D = \{8,9,10,11,12,13,14,15\}$; Group 6: $m=10$ $\cdot n=5$ $\cdot D = \{10,11,12,13,14,15\}$;

Group 7: $m=50$ $\cdot n=3$ $\cdot D = \{6,8,9,10,11,12,13,14\}$; Group 8: $m=50$ $\cdot n=4$ $\cdot D = \{8,9,10,11,12,13,14\}$;

Group 9: $m=50$ $n=5$ $D=\{10,11,12,13,14,15\}$; Group 10: $m=100$ $n=3$ $D=\{6,8,9,10,11,12,13\}$;
Group 11: $m=100$ $n=4$ $D=\{8,9,10,11,12,13,14\}$; Group 12: $m=100$ $n=5$ $D=\{10,11,12,13,14,15\}$.

Table 1 shows the results of running the program. The 3rd first columns, show the problem. First column is number of machines, second is number of jobs and third is number of sequence positions. The next three columns show average run-time of B&B algorithm to optimal solving with different amounts of α for 10 same problems. The next column shows average run time without parameter α . The next columns show average percent of deviations of makespan(ms), smoothing index(si), algorithm performing time(t) obtained by B&B with three value of parameter α from solving the model without parameter α . The percent of deviation is being computed as below for each of them:

$$\text{percent of deviation} = \frac{\text{B\&B solving result with } \alpha - \text{B\&B solving result without } \alpha}{\text{B\&B solving result without } \alpha}$$

The results show that with some percent (less than 10%) worse makespan for different amounts of α rather than the case without α , production smoothing index has improved significantly (at least 30 percent). For α_2 and α_3 makespan has been worse up to 3% but production smoothing index has improved about of 50%. Run-time of algorithm has improved about 50% that isn't so effective to solve larger problems ($D>11$) and it contributes to solve a problem with at most one more position rather than the case without α .

In case of $\alpha = \alpha_1$, makespan has been worse up to 10% but smoothing index has improved about 80%. Also in this case, α parameter with omitting more branches has a significant effect on algorithm run-time so that improves it up to 99%. This contributes to solve some problems with 3 or 4 more positions (D) in limited algorithm run-time.

Since number of problems has solved in case $\alpha = \alpha_1$ are greater than the three other cases, the problems has solved by heuristic algorithm and the results have compared.

Table 3 first shows average percent of deviation of heuristic solving from B&B solving without α and then average percent of deviation of B&B solving with different amounts of α from heuristic algorithm solving.

Results in Table 3 show that by increasing number of machines ($m = 50$ and 100) the deviation of heuristic algorithm makespan has increased significantly rather than the case without α parameter.

Results of Tables 2, 3 show by increasing number of machines in larger problems, makespan deviation of B&B solving with $\alpha = \alpha_1$ of B&B solving without α , is being decreased. In cases $m = 50$ and 100 this deviation equals to zero although smoothing index has improved between 60% to 80% and algorithm run-time has improved up to 99%.

Table 1. Results of B&B algorithm

M	N	D	BB1 Time	BB2 Time	BB3 Time	BB Opt Time	bb1, opt(ms)	bb1, opt(si)	bb1, opt(t)	bb2, opt(ms)	bb2, opt(si)	bb2, opt(t)	bb3, opt(ms)	bb3, opt(si)	bb3, opt(t)
3	3	6	0.00	0.02	0.03	0.03	5.11	-63.64	-85.00	2.94	-43.88	-35.00	2.90	-36.51	-20.00
3	3	8	0.04	1.09	1.46	1.68	6.09	-71.64	-97.45	2.61	-51.09	-34.47	2.60	-41.48	-12.77
3	3	9	0.28	10.31	13.14	14.66	8.63	-74.21	-98.27	2.08	-47.59	-29.28	2.02	-40.35	-10.08
3	3	10	1.02	120.02	142.99	153.72	7.43	-76.98	-99.30	2.65	-55.78	-21.13	2.21	-41.31	-6.65
3	3	11	4.49	300.02	300.02	300.02									
3	4	8	0.04	0.87	1.25	1.69	5.60	-65.55	-97.07	2.28	-46.03	-48.01	2.01	-40.55	-25.25
3	4	9	0.14	8.56	12.73	15.72	7.65	-68.43	-99.09	2.19	-48.55	-44.93	1.83	-40.40	-18.33
3	4	10	1.26	102.39	132.21	158.75	9.39	-71.09	-99.05	2.23	-49.57	-34.60	1.87	-40.86	-15.54
3	4	11	3.40	300.02	300.02	300.02									
3	5	10	2.16	80.75	117.15	152.29	6.95	-62.97	-98.41	1.91	-46.38	-46.20	1.42	-32.92	-22.38
3	5	11	4.90	300.02	300.02	300.02									
10	3	6	0.01	0.03	0.03	0.04	5.60	-60.86	-76.67	3.15	-39.61	-25.83	3.03	-31.28	-21.67
10	3	8	0.07	1.62	2.08	2.23	6.41	-72.56	-96.69	3.05	-51.03	-25.71	2.92	-43.11	-4.74
10	3	9	0.93	15.60	19.04	19.31	10.06	-76.37	-95.61	2.86	-50.90	-16.71	2.76	-44.43	1.95
10	3	10	2.05	152.12	189.34	181.62	6.36	-78.37	-98.72	2.76	-52.76	-13.17	2.71	-45.76	8.50
10	3	11	7.10	300.02	300.02	300.02									
10	4	8	0.06	1.31	1.73	1.99	6.87	-67.12	-96.81	3.88	-47.23	-29.16	2.91	-42.36	-7.48
10	4	9	0.19	13.06	16.69	17.58	9.37	-70.09	-98.86	3.70	-46.97	-19.99	2.57	-35.64	1.22
10	4	10	1.08	156.57	189.06	190.50	6.92	-71.86	-99.35	3.37	-49.58	-14.72	2.15	-35.24	2.52
10	4	11	3.59	300.02	300.02	300.02									
10	5	10	0.89	107.76	157.79	200.65	5.75	-67.53	-99.46	2.40	-48.93	-45.20	1.72	-40.35	-20.91
10	5	11	2.78	300.02	300.02	300.02									
50	3	6	0.04	0.12	0.13	0.18	0.00	-59.74	-78.48	0.00	-42.41	-35.61	0.00	-40.98	-27.80
50	3	8	0.26	6.20	8.26	9.91	0.00	-70.34	-97.37	0.00	-50.20	-37.50	0.00	-41.84	-16.68
50	3	9	4.19	64.15	83.78	96.06	0.00	-73.91	-95.34	0.00	-53.75	-32.91	0.00	-40.88	-10.99

M	N	D	BB1 Time	BB2 Time	BB3 Time	BB Opt Time	bb1, opt(ms)	bb1, opt(si)	bb1, opt(t)	bb2, opt(ms)	bb2, opt(si)	bb2, opt(t)	bb3, opt(ms)	bb3, opt(si)	bb3, opt(t)
50	3	10	9.18	300.02	300.02	300.02									
50	4	8	0.40	5.42	7.37	9.95	0.00	-64.14	-95.93	0.00	-46.58	-45.53	0.00	-36.82	-25.90
50	4	9	0.67	51.47	72.12	90.67	0.00	-69.77	-99.26	0.00	-47.15	-43.23	0.00	-36.00	-20.45
50	4	10	4.77	300.02	300.02	300.02									
50	5	10	0.89	107.76	157.79	200.65	5.75	-67.53	-99.46	2.40	-48.93	-45.20	1.72	-40.35	-20.91
50	5	11	10.59	300.02	300.02	300.02									
100	3	6	0.05	0.21	0.23	0.32	0.00	-64.13	-84.00	0.00	-48.65	-34.71	0.00	-43.73	-27.42
100	3	8	0.46	11.62	15.41	18.46	0.00	-70.72	-97.50	0.00	-49.03	-37.06	0.00	-44.73	-16.49
100	3	9	6.20	115.15	144.93	167.58	0.00	-74.51	-96.31	0.00	-51.43	-31.27	0.00	-38.00	-13.52
100	3	10	9.41	300.02	300.02	300.02									
100	4	8	0.50	9.72	13.54	18.65	0.00	-64.75	-97.32	0.00	-46.75	-47.85	0.00	-33.69	-27.42
100	4	9	1.33	93.05	131.90	168.15	0.00	-69.07	-99.21	0.00	-49.90	-44.64	0.00	-34.49	-21.54
100	4	10	21.20	300.02	300.02	300.03									
100	5	10	6.89	300.02	300.02	300.02									

Table 2. Comparison of B&B algorithm results with heuristic algorithm results

M	N	D	heu,opt(ms)	heu,opt(si)	bb1,heu(ms)	bb1,heu(si)	bb2,heu(ms)	bb2,heu(si)	bb3,heu(ms)	bb3,heu(si)
3	3	6	3.99	-10.80	1.13	-56.88	-0.97	-31.59	-1.01	-25.53
3	3	8	4.07	-19.28	1.96	-62.56	-1.37	-35.58	-1.38	-22.58
3	3	9	3.25	-0.22	5.21	-73.58	-1.13	-46.48	-1.19	-39.06
3	3	10	3.74	-17.65	3.56	-69.83	-1.01	-37.25	-1.43	-16.94
3	3	11	2.45	-18.43	4.78	-74.35	-0.91	-38.85	-1.10	-25.92
3	4	8	5.03	-11.27	0.55	-57.26	-2.60	-32.63	-2.84	-26.50
3	4	9	4.47	-13.02	3.02	-59.87	-2.17	-34.22	-2.51	-23.75
3	4	10	4.05	-13.83	5.11	-63.48	-1.73	-38.00	-2.07	-26.44
3	4	11	4.32	-14.93	2.76	-64.53	-2.31	-38.19	-2.62	-22.02
3	5	10	3.56	4.66	3.28	-63.76	-1.57	-47.90	-2.04	-35.49
3	5	11	4.16	1.42	1.85	-65.48	-2.86	-46.86	-3.17	-35.76
10	3	6	4.35	4.37	1.26	-60.35	-1.04	-38.68	-1.16	-32.02
10	3	8	5.14	-7.76	1.24	-68.53	-1.94	-44.32	-2.05	-35.33
10	3	9	4.80	-11.61	5.11	-72.15	-1.81	-42.72	-1.89	-34.70
10	3	10	4.77	-18.78	1.56	-71.30	-1.88	-38.54	-1.93	-28.87
10	3	11	2.97	-14.93	5.43	-73.84	-1.53	-40.32	-1.65	-27.97
10	4	8	3.86	-8.98	2.95	-59.90	0.06	-34.91	-0.85	-27.77
10	4	9	4.66	-2.36	4.55	-65.94	-0.86	-39.41	-1.91	-25.91
10	4	10	6.33	7.54	0.59	-70.58	-2.75	-47.97	-3.89	-34.79
10	4	11	5.12	-12.28	2.55	-69.22	-2.68	-41.57	-3.22	-29.78
10	5	10	4.78	-12.75	0.96	-59.99	-2.24	-36.70	-2.85	-26.69
10	5	11	-99.60	22474.41	-53.38	-99.69	44663.94	-99.77	-3.30	-17.95
50	3	6	-92.77	-7.46	2919.07	-53.95	2919.07	-33.51	2919.07	-32.02
50	3	8	-97.19	-4.30	3462.18	-67.86	3462.18	-46.13	3462.17	-38.00
50	3	9	-97.18	-7.52	3453.07	-69.87	3453.07	-46.81	3453.06	-30.39
50	3	10	-97.20	-13.47	3471.70	-70.18	3471.69	-37.37	3471.69	-28.15
50	4	8	-97.25	-8.34	3546.55	-60.28	3546.55	-40.18	3546.55	-30.23
50	4	9	-97.26	-11.05	3548.58	-64.59	3548.58	-37.09	3548.58	-22.47
50	4	10	-97.28	-20.34	3577.34	-63.00	3577.34	-31.36	3577.33	-22.48
50	5	10	4.78	-12.75	0.96	-59.99	-2.24	-36.70	-2.85	-26.69
50	5	11	-97.33	8.71	3643.96	-62.05	3643.96	-33.15	3643.96	-19.11
100	3	6	-98.76	-10.56	8175.91	-55.39	8175.91	-37.47	8175.90	-32.99
100	3	8	-91.35	-11.32	7425.43	-62.16	7425.43	-34.73	7425.43	-30.53
100	3	9	-91.29	-15.77	7275.84	-63.73	7275.84	-28.55	7275.84	-8.71
100	3	10	-98.83	-15.08	8482.92	-63.94	8482.92	-23.23	8482.91	-14.42
100	4	8	-98.83	-9.70	8492.31	-57.11	8492.31	-35.43	8492.30	-18.09
100	4	9	-98.63	-11.16	8037.50	-59.52	8037.50	-33.62	8037.50	-14.60
100	4	10	-98.66	-21.19	8040.11	-58.66	8040.11	-33.01	8040.10	-12.57
100	5	10	-98.53	-5.39	7427.94	-60.02	7427.94	-43.55	7427.94	-34.63

Table 4. Comparison of results for deferent group of problems has reached to optimal solving in 5 minutes.

α_3 with opt*			α_2 with opt			α_1 with opt			problem group
Time	smoothing index	makespan	time	smoothing index	makespan	time	smoothing index	makespan	
6 to 20% better	36 to 43% better	2 to 3% worse	21 to 35% better	43 to 56% better	2 to 3% worse	85 to 99% better	63 to 77% better	5 to 7.5% worse	1
15 to 25% better	40.4 to 40.9% better	1.8 to 2% worse	34 to 48% better	46 to 49% better	2.2 to 2.3% worse	97 to 99% better	65 to 71% better	5 to 9.4% worse	2
22.4% better	33% better	1.4% worse	46.2% better	46.4% better	1.9% worse	98% better	63 % better	6.95% worse	3
21 to 8.5% better	31 to 45.8% better	2.7 to 3.3% worse	13 to 25.8% better	39.6 to 53% better	2.8 to 3.15% worse	76 to 99% better	60 to 78% better	5.6 to 10% worse	4
7.4% better to 42.4% worse	35 to 42.4% better	2.15 to 2.9% worse	14.7 to 29.1% better	47 to 49.6% better	3.4 to 3.9% worse	96.8 to 99.4% better	67 to 72% better	6.1 to 9.4% worse	5
20.9% better	40.4% better	1.7% worse	45% better	49% better	2.4% worse	99.5 better	67.5% better	5.7% worse	6
10 to 27.8% better	40.9 to 41.9% better	equal	32.9 to 37.5% better	42.4 to 53.8% better	equal	78.5 to 95.4% better	59.7 to 73.9% better	equal	7
20.4 to 25.9% better	36 to 36.8% better	equal	43 to 45.5% better	46.6 to 47% better	equal	96 to 99% better	64 to 69.7% better	equal	8
20.9% better	40.4% better	1.7% worse	45.2% better	48.9% better	2.4% worse	99.5 better	67.5% better	5.75% worse	9
13.5 to 27.4% better	38 to 46.7% better	equal	31.3 to 37% better	48.6 to 51.4% better	equal	84 to 97.5% better	64 to 74.5% better	equal	10
21.5 to 27.4% better	33.7 to 34.5% better	equal	46.6 to 47.9% better	46.7 to 49.9% better	equal	97.3 to 99.2% better	64.7 to 69% better	equal	11

* B&B algorithm solving without α

5. Conclusions

In this paper a model and an exact method for solving the model based on branch and bound and also a heuristic algorithm is proposed to minimizing the makespan in the Flow shop scheduling problem with sequence dependent set-up times with considering production smoothing. The model and solving methods are not reported in the literature. Due to The proposed exact method (B&B) can not solve large scale problems this is an opportunity for future researches. Improving the proposed B&B by define efficient lower bound and also using a meta- heuristic method can be considered in future researches. Considering other production environments like Hybrid flow shop is another opportunity for future researches.

References

1. Hejazi, S.R., Saghaian, S, "Flowshop-scheduling problems with makespan criterion: a review", 2005, International Journal of Production Research, 43, 2895–2929.
2. Yavuz, M., Akçali, E., 2007, "Production smoothing in just-in-time manufacturing systems: a review of the models and solution approaches", International Journal of Production Research, 45, 3579 – 3597,.
3. McMullen, P.R. 2002 "The permutation flow shop problem with just in time production considerations", Production Planning & Control, 13(3), 307-316,
4. Yavuz, M., Akçali, E., Tufekci, S., 2006, "A hybrid meta-heuristic for the batching problem in just-in-time flow shops.", Journal Math. Model. Algor., 5, 371–393.
5. Yavuz, M., Tufekci, S., 2007, "Dynamic programming solution to the batching problem in just in-time flowshops.", Computers & Industrial Engineering 51, 416–432.
6. Kogan, K., Tell, H., 2009, " Production smoothing by balancing capacity utilization and advance orders", IIE Transactions, 41(3), 223-231
7. Yalcinkaya, A., Dumrmusoglu, M.B., " Production smoothing decisions with seasonal variation through value stream management, 2009, ", Proc. Of International conference in Industrial Eng. Conf. , 6-9 July, Troyes, 128-133