Scheduling Two-machine Flowshop with Limited Waiting Times to Minimize Makespan

Bailin Wang*[1,2], Tieke Li[1,2], Cantao Shi[1,2], Haifeng Wang[1,2]
[1]Dongling School of Economics and Management; University of Science and Technology Beijing;
Beijing 100083; China
[2]Engineering Research Center of MES Technology for Iron & Steel Production; Ministry of Education;
Beijing 100083; China
*Corresponding author, e-mail: wangbailin_83@sina.com

Abstract

There are numerous instances of flowshop in the production process of process industry. When such characteristics as continuous production resulted from high-temperature environment or deteriorate intermediate products are took into consideration, it should be ensured that the waiting time of any job between two consecutive machines is not greater than a given value, which results in the flowshop scheduling problem with limited waiting time constraints. The problem with two-machine environment to minimize makespan is studied. Based on the discussion of the lower bound of the minimal makespan and some properties of the optimal schedule, a two-stage search algorithm is proposed, in which the initial schedule is generated by a modified LK heuristic in the first stage and the excellent solution can be obtained by constructing inserting neighborhood in the second stage. The numerical results demonstrate the effectiveness of the algorithm.

Keywords*: scheduling, two-machine flowshop, limited waiting times, heuristic*

1. Introduction

Flowshop scheduling problem is a class of combinatorial optimization problem with a strong background of engineering. In a flowshop, there is a set of jobs to be processed on a number of machines in series, and all jobs have to follow the same route. In some manufacture with a continuous production characteristics resulted from high-temperature environment or deteriorate intermediate products, such as steelmaking production [1] and food production [2], it should be ensured that the waiting time of any job between two consecutive machines must not be greater than a given value, which results in a flowshop scheduling problem with limited waiting time constraints.

There are only a few studies of the flowshop scheduling with limited waiting time constraints. For the problem with two machines, Yang and Chern [3] proved that the problem is NP-hard, and proposed a branch-and-bound algorithm; Wang and Li [4] further proved that the problem is strongly NP-hard, and extended classical two-machine flowshop scheduling heuristics for comparative analysis; Su [5] presented a heuristic algorithm for a hybrid two-stage flowshop with a batch processor in stage 1 and a single processor in stage 2; Joo and Kim [6] analyzed some problem characteristics, and suggested a branch-and-bound algorithm for the problem. For the multi-stage flowshop environment, Chen and Yang [7] studied the modeling mechanism of scheduling problem with limited waiting time constraints; Wang and Li [8] discussed the characters of job sequences on machines, and analyzed the feasibility and validity of problem-solving approaches concerning permutation schedules; Wang and Li [9] presented some flowshop scheduling heuristics for limited waiting time constraints, and carried out a comparison of those heuristics; Dhouib et al. [10] considered the objectives of the minimization of the number of tardy jobs and makespan, and propose a simulated annealing algorithm to heuristically solve the problem.

Two-machine production scheduling with the makespan objective is a foundation problem in the flowshop scheduling domain. Previous researches mainly adopt branch-and-bound algorithms [3, 6] or heuristic extension algorithms [4] for the two-machine flowshop scheduling with limited waiting times. However, the former solve large-scale problem

Received August 28, 2013; Revised November 16, 2013; Accepted December 7, 2013

inefficiently because of the exponential growth of the calculated time and storage space, and the later cannot effectively obtain optimal solution without applying the problem characteristics. This paper attends to analyze the characteristics of job sequence and explore a two-stage search heuristic algorithm for the problem to minimize makespan. The rest of the paper is organized as follows. Section 2 describes the scheduling problem and analyzes its lower bound. A heuristic algorithm is presented in Section 3, and its performance is verified by computational experiments in Section 4. Finally, Section 5 concludes the paper with a short summary.

2. The Problem and its Lower Bound

A two-machine flowshop scheduling problem with limited waiting time constraints can be formulated as follows. There are n jobs $\{J_1, J_2, \ldots, J_n\}$ which have to be processed on two machines $\{M_1, M_2\}$ via the same route. The processing time of J_i on M_j is known and denoted as p_{ij}. The problem is to find a job sequence π and determine the completion time C_{ij} of job J_i $(i = 1, 2, \cdots, n)$ on machine M_j $(j = 1, 2)$ to satisfy the following constraints.

Constraint 1: Each job should be processed on M_2 after completion on M_1 (Equation (1)).

$$C_{i2} - C_{i1} \geq p_{i2} \quad (i = 1, 2, \cdots, n) \tag{1}$$

Constraint 2: Each machine may start to process one job after the previous one has completed, which can be expressed as Equation (2) where $\pi(k)$ is the *k*-th processing job on a sequence π.

$$C_{\pi(k),j} - C_{\pi(i),j} - p_{\pi(k),j} \geq 0 \quad (1 \leq i < k \leq n; j = 1, 2) \tag{2}$$

Constraint 3 (Limited Waiting Time Constraints): The waiting times of J_i between two consecutive machines M_1 and M_2, denoted as w_i, must not be greater than a given upper bound α (Equation (3)).

$$w_i = C_{i2} - C_{i1} - p_{i2} \leq \alpha \quad (i = 1, 2, \cdots, n) \tag{3}$$

The optimization objective is to acquire an optimal job sequence π^* and completion times $\{C_{ij} \mid i = 1, 2, \cdots, n; j = 1, 2\}$ to minimize the makespan as Equation (4) shown.

$$\min C_{\max} = \max_{i,j} \{C_{ij}\} \tag{4}$$

Using the notation proposed by Graham et al. [11], the problem can be denoted as $\mathrm{F2}|w_i \leq \alpha|C_{\max}$.

While the upper bound of waiting time approaches infinite ($\alpha \to \infty$), the problem is equal to the general two-machine flowshop scheduling problem $\mathrm{F2}| \, |C_{\max}$, which can be solved by Johnson's rule in polynomial time $O(n \lg n)$. Let $C_{\max}(OPT)$ and $C_{\max}(gnr)$ denotes the optimal objective values of $\mathrm{F2}|w_i \leq \alpha|C_{\max}$ and its corresponding problem $\mathrm{F2}| \, |C_{\max}$ respectively, it is interesting to further discuss a relationship between them.

Suppose π^* is an optimal job sequence of $\mathrm{F2}|w_i \leq \alpha|C_{\max}$. It is clear that π^* is also a feasible sequence of $\mathrm{F2}| \, |C_{\max}$. The optimal schedule of $\mathrm{F2}|w_i \leq \alpha|C_{\max}$ can be obtained by right shifting the completion time of jobs which dissatisfy the limited waiting time constraints in π^* of $\mathrm{F2}| \, |C_{\max}$, as shown in Figure 1.

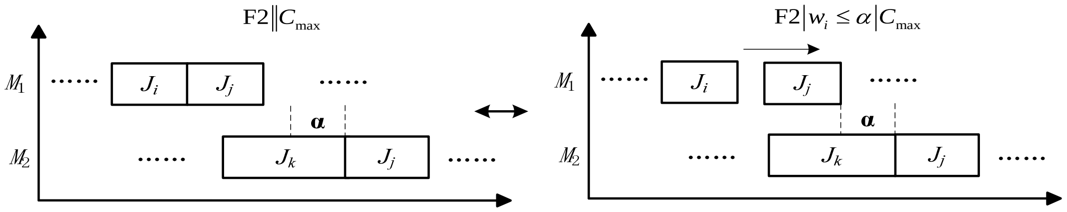

Figure 1. The Schedules of $\text{F2}\|C_{\max}$ and $\text{F2}|w_i \le \alpha|C_{\max}$

Thus we have $C_{\max}^{lwt}(\pi^*) \ge C_{\max}^{gnr}(\pi^*)$, where $C_{\max}^{lwt}(\pi^*)$ and $C_{\max}^{gnr}(\pi^*)$ denote the makespans of schedules with sequence π^* in $\text{F2}|w_i \le \alpha|C_{\max}$ and $\text{F2}\|C_{\max}$ respectively. Because $C_{\max}^{lwt}(\pi^*) = C_{\max}(OPT)$ and $C_{\max}(\pi') \ge C_{\max}(gnr)$, the following formula Eq.5 can be acquired.

$$C_{\max}(OPT) \ge C_{\max}(gnr) \tag{5}$$

Therefore, the value of $C_{\max}(gnr)$, which may be calculated by Johnson's rule in polynomial time $O(n\lg n)$, can be applied as a lower bound of $\text{F2}|w_i \le \alpha|C_{\max}$.

3. Two-Stage Search Algorithm

3.1. Main Idea for Solving the Problem

Problem $\text{F2}|w_i \le \alpha|C_{\max}$ can be decomposed into two subproblems, job sequencing and time variable assignment. For the later, a feasible schedule of any job sequence can be obtained by delaying the completion time of the jobs on M_2 of which waiting times are greater than the upper bound, as shown in Figure 2.

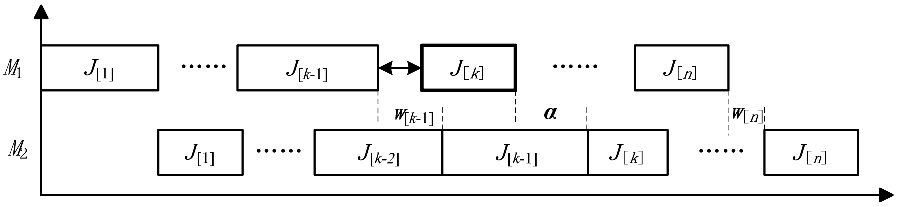

Figure 2. Approach to Calculate Completion Times of Jobs

The completion time of each job in a sequence π may be calculated as Equation (6)-Equation (9) shown.

For the first processing job $J_{\pi[1]}$:

$$C_{\pi[1],1} = p_{\pi[1],1} \tag{6}$$

$$C_{\pi[1],2} = p_{\pi[1],1} + p_{\pi[1],2} \tag{7}$$

For other jobs $J_{\pi[k]}$ where $k > 1$:

$$C_{\pi[k],1} = \max\left\{C_{\pi[k-1],1} + p_{\pi[k],1}, C_{\pi[k-1],2} - \alpha\right\} \tag{8}$$

$$C_{\pi[k],2} = \max\left\{C_{\pi[k],1}, C_{\pi[k-1],2}\right\} + p_{\pi[k],2} \tag{9}$$

Thus the former subproblem, job sequencing, is the key challenge for the two-machine flowshop scheduling with limited waiting times. An interesting property of the minimum makespan present by Wang and Li [4] and shown as Equation (10) may provide some thought to sequence jobs.

$$\min C_{\max} < \frac{1}{2}\left[\sum_{i=1}^{n}\sum_{j=1}^{2} p_{ij} + \min_{\pi}\left\{p_{\pi[1],1} + \sum\nolimits_{k=2}^{n}\left|p_{\pi[k],1} - p_{\pi[k-1],2}\right| + p_{\pi[n],2}\right\} + \alpha\right] \tag{10}$$

Set $z_\pi = p_{\pi[1],1} + \sum\nolimits_{k=2}^{n}\left|p_{\pi[k],1} - p_{\pi[k-1],2}\right| + p_{\pi[n],2}$, and then we have $\min C_{\max} < \frac{1}{2}\left[\min_{\pi}\{z_\pi\} + \sum_{i=1}^{n}\sum_{j=1}^{2} p_{ij} + \alpha\right]$. In the right side of the inequality, only $\min_{\pi}\{z_\pi\}$ is related to the job sequence. Hence, we suggest an idea for job sequencing that if π^* is an optimal sequence satisfying $\pi^* \in \left\{\pi \left| \min_{\pi}\{z_\pi\}\right.\right\}$, then the corresponding schedule may approach or even reach the optimal solution of the problem.

The character of $\min_{\pi}\{z_\pi\}$ should be further analyzed here. If a virtual job J_0 with $p_{01} = p_{02} = 0$ is added to the job set as $\{J_0, J_1, J_2, \ldots, J_n\}$, $\min_{\pi}\{z_\pi\}$ is equivalent to $\min\ z'_\pi = \sum\nolimits_{k=1}^{n}\left|p_{\pi[k-1],2} - p_{\pi[k],1}\right|$, which is a formalize model of Travelling Salesman Problem (TSP) where each job J_i is the equal of a city, and the distance between two cities J_i and J_j is $d_{ij} = \left|p_{i2} - p_{j1}\right|$. Obviously the problem to find $\min_{\pi}\{z_\pi\}$ is an asymmetry TSP because $d_{ij} \neq d_{ji}, \exists i, j = 1, 2, \cdots, n$, which is one of the most difficult travelling salesman problem.

Therefore, it is a way to obtain a satisfactory schedule of $\mathrm{F2}\left|w_i \leq \alpha\right|C_{\max}$ by searching for an optimal job sequence of the asymmetry TSP $\min_{\pi}\ z'_\pi = \sum\nolimits_{k=1}^{n}\left|p_{\pi[k-1],2} - p_{\pi[k],1}\right|$. Based on the idea, we propose a two-stage search algorithm that the first stage is to obtain an initial solution with the consideration of the makespan character and the second stage is to search for a better schedule by constructing an inserting neighborhood.

3.2. Stage 1: Modified LK Heuristic

TSP is a NP-hard combinational optimization problem, and there are a mass of methods for the problem, such as branch-and-bound algorithms, heuristics, genetic algorithms, neighborhood search algorithms, etc. The Lin-Kernighan heuristic (short for LK heuristic) proposed by Lin and Kernighan in 1973 [12] is a variable *k*-opt neighborhood search heuristic, which is generally considered to be one of the most effective methods for generating optimal or near-optimal solutions for the symmetric TSP [13].

Based on the solution idea, a *modified LK heuristic* (MLK for short) is proposed to get an initial solution. In this algorithm, an extended LK is embedded into the job sequencing process to obtain a good-enough solution of the TSP $\min_{\pi}\ z'_\pi = \sum\nolimits_{k=1}^{n}\left|p_{\pi[k-1],2} - p_{\pi[k],1}\right|$ by variable *k*-opt neighborhood search technique, and the schedule is determined by right shifting the start times of jobs in the sequence. One important thing to note about MLK heuristic is that $\min_{\pi}\ z'_\pi = \sum\nolimits_{k=1}^{n}\left|p_{\pi[k-1],2} - p_{\pi[k],1}\right|$ is an asymmetry TSP which cannot be solved directly by the traditional LK algorithm, thus we explore an extended LK method with a modified sequential exchange criterion.

The MLK heuritic in Stage1 is shown below.

Step 1 Initialization.

a) Set $iter=1$, $C^*_{\max}=U$ and $BestSch=\varnothing$. $MaxIter$ is the maximum iterative number.

b) *Construct the corresponding TSP of problem* $\mathrm{F2}|w_i\le\alpha|C_{\max}$*:* Build a Graph $G=(V,A)$ where the city set $V=\{0,1,2,\dots,n\}$ and the distance matrix $D=\{d_{ij}=|p_{i2}-p_{j1}|\,|\,i\ne j\wedge i,j\in V\}$.

Step 2 Generating job sequence by the extended LK heuristic

a) Generate an initial tour randomly. Let $BestTour=Tour$.

b) Let $i=1$.

c) In consideration of the asymmetry of the constructed TSP, the edge selection rule shown below is different from the general LK algorithm.

Edge Selection Rule: Select $x_i=(t_{2i-1},t_{2i})\in Tour$ and $y_i=(t_{2i+1},t_{2i})\notin Tour$ which would get a better tour by the exchange of $\{x_1,\dots,x_i\}$ and $\{y_1,\dots,y_i\}$.

d) If it cannot obtain a better solution, then go to Step2e; else, set $i=i+1$, go to Step2c.

e) If the best improvement is reached when $k=i$, generate a new tour named $Tour$ by *k*-opt, let $BestTour=Tour$, and go to Step2b. If there is no more improvement, translate the set $BestTour$ into a node set $BestNode$ which means the job sequence of the scheduling problem, and then go to Step3.

Step 3 Calculating the completion times of jobs and makespan in the sequence.

a) Set $k=1$, $J^{(k)}=J_{BestNode(k)}$, $C_{(k)1}=p_{(k)1}$, and $C_{(k)2}=C_{(k)1}+p_{(k)2}$.

b) Set $k=k+1$. If $k\le n$ then go to Step3c, else set $C_{\max}=C_{(n)2}$. If $C_{\max}<C^*_{\max}$,set $C^*_{\max}=C_{\max}$ and $BestSch=BestNode$, then go to Step4.

c) Set $J^{(k)}=J_{BestNode(k)}$. Calculate the right shift value $d^*_{(k)}$ as Equation (11) shown.

$$d^*_{(k)}=\max\{0,p_{(k-1)2}-p_{(k)1}+w_{(k-1)}-\alpha\} \tag{11}$$

Set $C_{(k)1}=C_{(k-1)1}+d^*_{(k)}+p_{(k)1}$, $C_{(k)2}=\max\{C_{(k)1},C_{(k-1)2}\}+p_{(k)2}$. Then go to Step3b.

Step 4 Termination Criterion

Set $iter=iter+1$. If $iter\le MaxIter$ then go to Step 2, else MLK algorithm is terminated and output the final schedule.

3.3. Stage 2: Inserting Neighborhood Heuristic

To further optimize the initial schedule generated in Stage1, an *inserting neighborhood heuristic* (INH for short) is proposed with a constructive neighborhood defined.

Definition 1 (J_i *-neighborhood)* A schedule-set, denoted as $Ins(\pi,J_i)$, is called J_i-*neighborhood of Schedule* π while it is composed of schedules in which the positions of all the jobs except J_i are the same as π**.**

J_i-neighborhood of Schedule π is composed of $n-1$ schedules which may be obtained as follows. Firstly, generate a subsequence $Seq(\pi-J_i)$ by removing J_i from π. Secondly, insert J_i into the $n-1$ positions which are not the position of job J_i in π respectively to get $n-1$ new schedules. These schedules compose the J_i-neighborhood of π.

For example, if $n=5$ and $\pi=\{J_2,J_5,J_3,J_1,J_4\}$, we have $Seq(\pi-J_3)=\{J_2,J_5,J_1,J_4\}$, and the J_3-neighborhood of π is shown below.

$$Ins(\pi,J_3)=\{\{J_3,J_2,J_5,J_1,J_4\},\{J_2,J_3,J_5,J_1,J_4\},\{J_2,J_5,J_1,J_3,J_4\},\{J_2,J_5,J_1,J_4,J_3\}\}$$

The search process of INH, which depends on the job sequence of the initial schedule denoted as $\pi_0=\{J^0_{[1]},\dots,J^0_{[n]}\}$, is firstly to generate $Ins(\pi_i,J^0_{[i]})$ as the $J^0_{[i]}$-neighborhood of current best solution π_i where $i\ (i=1,\ldots,n)$ is the current iteration number, and then to search a better schedule from the schedule set $Ins\left(\pi_i, J_{[i]}^0\right)$.

The INH heuritic in Stage 2 is shown below.

Step 5 Get the initial schedule $\pi_0=\left\{J_{[1]}^0,\ldots,J_{[n]}^0\right\}$ from Stage1. Set current best sequence $\pi^*=\pi_0$ and current best objective value $C_{\max}^*=C_{\max}^0$. Set the number of iteration $i=1$;

Step 6 Set $\pi_i=\pi^*$, $k=1$, $k^*=-1$. Let k_0 denote the position of $J_{[i]}^0$ in π_i. Generate a subsequence $Seq\left(\pi_i-J_{[i]}^0\right)$;

Step 7 If $k=k_0$, go to Step8; else insert $J_{[i]}^0$ into the k-position of $Seq\left(\pi_i-J_{[i]}^0\right)$ and mark as $Ins\left(\pi^i, J_{[i]}^0\right)[k]$, and calculate completion times of jobs and the makespan $C_{\max}$ by Equation (6)-Equation (9). If $C_{\max}<C_{\max}^*$, set $C_{\max}^*=C_{\max}$ and $k^*=k$;

Step 8 Set $k=k+1$. If $k\le n$, go to Step7, else set $\pi^*=Ins\left(\pi^i, J_{[i]}^0\right)\left[k^*\right]$;

Step 9 Set $i=i+1$. If $i\le n$, go to Step6, else output π^* and $C_{\max}^*$.

3.4. Time Complexity of the Algorithm

The above algorithm is executed by two stages, Modified LK Heuristic and Inserting Neighborhood Heuristic.

It can be seen from Stage1 that the complexity to calculate distances between any two nodes in Step1 is $O\left(n^2\right)$. The time complexity of LK in Step2 is approximately $O\left(n^{2.2}\right)$[**Error! Bookmark not defined.**]. Step 3 to calculate completion times of jobs should be in $O(n)$ time. Step 4 is independent of n. Therefore, the whole MLK takes $O\left(n^{2.2}\right)$ to compute.

In Stage 2, Step 6-Step 8 form an inner loop with $n-1$ iteration, and the time complexity is $O\left(n^2\right)$ because calculating time variables of a schedule would take $O(n)$. Step 5-Step 9 is the outer loop with n iteration. Therefore, INH may be computed in $O\left(n^3\right)$ time.

Therefore, the Two-Stage Search Algorithm takes $O\left(n^3\right)$ to compute.

4. Computational Experiments

In this section, we carry out computational experiments to verify the effectiveness of the Two-Stage Search Algorithm by comparing with that of extended Johnson's rule [**Error! Bookmark not defined.**] and MLK Hueristic. The heuristics are coded in C# language and implemented on the computer with Core i5/ CPU32.4GHz/ RAM3.0G. Set the maximum iterative number of Stage 1 in the Two-Stage Search Algorithm and MLK Hueristic is 1000, and the termination criterion is that the iterative number reaches the maximum value or the makespan of the best schedule is equal to the lower bound calculated by Johnson's rule.

The main factors affecting the performance of algorithms are problem size (the number of jobs) and the upper bound of waiting time. Therefore, the parameters of test problems are grouped as Table 1 shown, and each group is divided further according to the interval length of α as 10 time unit with 20 random generated test problems. The performance is measured by the percentage deviation $D(H)$ where H is the name of the algorithm and $Lower=C_{\max}(gnr)$.

$$D(H)=\frac{C_{\max}(H)-Lower}{Lower}\times 100\% \tag{12}$$

Table 1. Parameters of Test Problems

Group No.	Job Number	Upper bound of waiting times	Processing Times
Group1	$n = 20$	$\alpha \in DU[0,50]$	$p_{ij} \in DU\left[10,30\right]$
Group2	$n = 50$	$\alpha \in DU[0,80]$	$p_{ij} \in DU\left[10,30\right]$
Group3	$n = 80$	$\alpha \in DU[0,120]$	$p_{ij} \in DU\left[10,30\right]$

Results are given in Table 2, where CPU is the mean computing time of the Two-Stage Search Algorithm (TSA for short).

Table 2. Computational Results

Group No.	No.	α	CPU(s)	$\overline{D(H)}$(%)		
				TSA	MLK	Johnson's
Group1 ($n = 20$)	1	$[0,10]$	1.616	0.53	8.20	9.44
	2	$[10,20]$	1.625	0.12	4.17	6.23
	3	$[20,30]$	1.661	0.01	3.47	2.67
	4	$[30,40]$	1.658	0.09	3.58	2.24
	5	$[40,50]$	1.643	0	3.23	0.35
Group2 ($n = 50$)	6	$[0,10]$	25.201	0.18	7.18	11.89
	7	$[10,20]$	24.581	0.02	2.85	10.89
	8	$[20,30]$	24.343	0	1.43	9.53
	9	$[30,40]$	23.950	0.01	1.17	8.89
	10	$[40,50]$	24.095	0	1.12	8.72
	11	$[50,60]$	24.115	0.02	1.19	6.65
	12	$[60,70]$	23.973	0	1.16	6.54
	13	$[70,80]$	24.76	0	1.21	6.10
Group3 ($n = 80$)	14	$[0,10]$	108.269	0.07	7.11	12.59
	15	$[10,20]$	104.224	0.04	4.01	12.88
	16	$[20,30]$	109.564	0	1.79	11.98
	17	$[30,40]$	108.946	0	1.53	12.12
	18	$[40,50]$	109.073	0	0.96	10.01
	19	$[50,60]$	108.244	0	0.88	10.00
	20	$[60,70]$	107.132	0	0.78	9.56
	21	$[70,80]$	105.726	0	0.75	8.18
	22	$[80,90]$	109.599	0	0.83	8.58
	23	$[90,100]$	108.517	0	0.79	7.53
	24	$[100,110]$	107.650	0	0.75	6.78
	25	$[110,120]$	109.338	0	0.79	5.76

The results demonstrate the following points.

Firstly, the performance of the Two-Stage Search Algorithm is obviously better than that of extended Johnson’s rule and Modified LK Heuristic. And the larger the problem size is or the tighter the limited waiting time constraints are, the better the optimization effect of TSA is. Moreover, the percentage deviations of TSA were much lower than that of MLK, which is used to get an initial schedule in TSA. It means that the optimization of Stage 2 is remarkable. And the percentage deviations of TSA reached zero in many cases, which indicates the lower bound present in this paper is tight, and these schedules obtained by TSA are the optimal solutions.

Secondly, with the increase of α, the percentage deviations of the three heuristics decline obviously. It shows that the effect of limited waiting time constraints on the problem solving is weakened gradually and lose binding force on more and more schedules while the upper bound of waiting times is rising.

Thirdly, from Table 2, it can be seen that to gain an good-enough schedule by extended Johnson's rule is hard while the job number is large. On the contrary, the percentage deviations of TSA and MLK continually decrease with the increase of *n*. This is because a large problem size will tighten the restrictions on jobs and make the problem complicated, and the performance of TSA and MLK which utilize the characters of waiting time restrictions in job sequencing, are not easily influenced by the problem size.

Finally, concerning the computational efficiency, the two-stage search algorithm can generate schedules in a short time to meet the requirement of production scheduling.

5. Conclusion

The flowshop with limited waiting time constraints is a common production environment in process industry. This paper studies the relevant scheduling problem with two machines to minimize makespan, which is strongly NP-hard that the optimal solutions cannot obtained in polynomial time. For solving the problem, a lower bound of the problem is given, and a two-stage search algorithm is presented. In the algorithm, the first stage is to obtain an initial solution by modifying LK heuristic with the consideration of the makespan character and the second stage is to search for a better schedule by constructing an inserting neighborhood. Experimental results shows the effectiveness and efficiencyof the algorithm, and meanwhile indicate that if the features of the special constraints of waiting times are taken into account in this process, the algorithm performance will be improved effectively.

.

Acknowledgements

This work was financially supported by the China Postdoctoral Science Foundation (2012M510324), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20100006110006), and the Fundamental Research Funds for the Central Universities (FRF-TP-12-116A and FRF-SD-12-012B).

References

[1] Pan QK, Wang L, Mao K, ect. An Effective Artificial Bee Colony Algorithm for a Real-World Hybrid Flowshop Problem in Steelmaking Process. *IEEE Transactions on Automation Science and Engineering*. 2013; 10: 307-322.

[2] Hodson A, Muhlemann AP, Price DHR. A Microcomputer based Solution to a Practical Scheduling Problem. *Operational Research Society*. 1985; 36(10): 903-914.

[3] Yang DL, Chern MS. A Two-Machine Flowshop Sequencing Problem with Limited Waiting Time Constraints. *Computers and Industrial Engineering*. 1995; 28(1): 63-70.

[4] Wang B, Li T. *Heuristics for Two-Machine Flowshop Scheduling with Limited Waiting Time Constraints*. Proceedings of 20th ICPR. China. 2009.

[5] Su LH. A Hybrid Two-Stage Flowshop with Limited Waiting Time Constraints. *Computers & Industrial Engineering*. 2003; 44: 409-424.

[6] Joo BJ, Kim YD. A Branch-and-bound Algorithm for a Two-Machine Flowshop Scheduling Problem with Limited Waiting Time Constraints. *Journal of the Operational Research Society*. 2009; 60(4): 572-582.

[7] Chen JS, Yang JS. Model Formulations for the Machine Scheduling Problem with Limited Waiting Time Constraints. *Journal of Information and Optimization Sciences*. 2006; 27: 225-240.

[8] Wang B, Li T. Some Properties of the Flowshop Scheduling Problem with Limited Waiting Time Constraints. *System Engineering-Theory & Practice*. 2012; 32: 1346-1352.

[9] Wang B, Li T. *A Comparison of Permutation Flowshop Scheduling Heuristics for Limited Waiting Time Constraints*. Proceedings of BMEI2011. China. 2011.

[10] Dhouib E, Teghem J, Loukil T. Lexicographic Optimization of a Permutation Flow Shop Scheduling Problem with Time Lag Constraints. *International Transactions in Operational Research*. 2013; 20: 213-232.

[11] Graham R, Lawler E, Lenstra J, et al. Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics.* 1979; 5: 287-326.

[12] Lin S, Kernighan BW. An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Operations Research*. 1973; 21: 498-516.
[13] Helsgaun K. An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. *European Journal of Operational Research.* 2000; 126: 106-130.