Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 395947, 9 pages
doi:10.1155/2012/395947

*Research Article*

# Optimal Results and Numerical Simulations for Flow Shop Scheduling Problems

**Tao Ren,[1] Yuandong Diao,[2] and Xiaochuan Luo[3]**

[1] *Software College, Northeastern University, Shenyang 110004, China*
[2] *Personnel Department, Shenyang University of Chemical Technology, Shenyang 110142, China*
[3] *State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110004, China*

Correspondence should be addressed to Tao Ren, chinarentao@163.com

Received 20 May 2011; Revised 26 October 2011; Accepted 4 January 2012

Academic Editor: Yuri Sotskov



This paper considers the $m$-machine flow shop problem with two objectives: makespan with release dates and total quadratic completion time, respectively. For $Fm|r_j|C_{max}$, we prove the asymptotic optimality for any dense scheduling when the problem scale is large enough. For $Fm\|\Sigma C_j^2$, improvement strategy with local search is presented to promote the performance of the classical SPT heuristic. At the end of the paper, simulations show the effectiveness of the improvement strategy.

## 1. Introduction

Industrial production is one of the most essential parts for the economic development with regard to its important value to economy and society in a country. Meanwhile, control and optimization are playing a vital role in promoting the productivity for industrial production. Whereas, the classical optimal methods are invalid and large-size computation is necessary because of the especial characteristic of discrete variables in industrial control and optimization problems. While the appearance of advanced algorithms and methods support for conquering these problems.

In this paper, two industrial optimization problems, flow shop scheduling problem to minimize makespan with release dates and the sum of quadratic completion times, are considered. In a flow shop model, there are a number of machines, and all jobs have to be processed on these machines following the constant route. A comprehensive overview about flow shop problem can be found in [1]. With the standard scheduling notation of Graham et al. [2], the flow shop scheduling problem to minimize makespan with release dates and the sum of quadratic completion times can be described by $\mathrm{F}m|r_j|C_{\max}$ and $\mathrm{F}m||\Sigma C_j^2$, respectively, where $m$ is the number of machines.

For $\mathrm{F}m|r_j|C_{\max}$, only a few researches can be found among the references. In 1977, Lenstra et al. [3] proved that $\mathrm{F}2|r_j|C_{\max}$ is strongly NP hard. The polynomial algorithm will be impossible to reach the optimal solution, if the class $P$ of problems is not equal to the class NP. Therefore, heuristics may be effective for large size problems to obtain an approximation solution. Some polynomial time approximation algorithms are investigated by Potts [4] for two-machine case. And the best algorithm among them has a worst case ratio of 5/3. Specially, Hall [5] presented a PATS (polynomial time approximation scheme) which is the strongest result known for the general problem.

Likewise, for $\mathrm{F}m||\Sigma C_j^2$, there are only few researchers that consider this quadratic objective, which may optimize the makespan and the total completion time together [6]. In 2005, Koulamas and Kyparisis [7] reported that $\mathrm{F}m||\Sigma C_j^2$ is strongly NP hard which implies that its optimal solution cannot be obtained in polynomial time. Therefore, they utilized the shortest processing time (SPT) heuristic to deal with large size problems and proved the asymptotic optimality and the worst case for the heuristic, respectively. Although the objective value of SPT heuristic equals the optimal solution as the number of jobs large enough, for some special case, the gap between them is the square of the number of machines.

For convenience of research, it is supposed that the jobs have bounded random (interval) processing times in flow shop problems typically. With the bounded uncertain processing times, [8] presented some dominant schedules for two-machine flow shop problem to minimize makespan and [9] obtained the necessary and sufficient conditions in a minimal set of dominant schedules for flow shop minimum-length scheduling problem with two machines. Under the hypothesis that the jobs have bounded independently and identically distributed (i.i.d.) processing times, [10] provided two lower bounds for $m$-machine flow shop problems, and [11] designed two heuristics for $m$-machine flow shop problems to minimize makespan.

In our work, we first study dense schedule for the flow shop makespan problem with release dates. Dense schedule was dealt with open shop makespan problem by Bárány and Fiala [12] first. In a dense schedule, any machine is idle if and only if there is no job that can be processed at that time on that machine. For problem $\mathrm{F}m||C_{\max}$, Chen and Yu [13] proved that the worst case performance ratio of any dense schedule is $m$, and the bound is tight. We show that any dense permutation schedule is asymptotically optimal for $\mathrm{F}m|r_j|C_{\max}$ as the number of jobs trends to infinity. Next, for $\mathrm{F}m||\Sigma C_j^2$, we try to present an improvement strategy based on local search to promote the performance of classical heuristic algorithm SPT. At the end of the paper, simulations show the effectiveness of the improvement strategy.

The remainder of the paper is organized as follows. The problem is formulated in Section 2, and the asymptotic optimality of dense schedule for $\mathrm{F}m|r_j|C_{\max}$ is in Section 3. In Sections 4 and 5, the improvement strategy and numerical simulations for $\mathrm{F}m||\Sigma C_j^2$ are given, respectively. And in Section 6, this paper is closed by the conclusions.

## 2. Problem Statement and Preliminaries

In a flow shop problem a set of $n$ jobs has to be sequentially processed on $m$ different machines. Each job $j$, $j = 1, 2, \ldots, n$, passes through the $m$ machines in that order and requires processing time $p(i, j)$ on machine $i$, $i = 1, 2, \ldots, m$, and a release date $r_j$. The processing times are i.i.d. bounded random variables. It is not permitted to process any job before its release date. At any given time each machine can handle at most one job and each job can be processed on at most one machine. Preemption is forbidden, that is, any commenced operation has to be completed without interruptions. And each machine processes the jobs in a first come first served manner. Here we consider the permutation schedule, that is, all jobs are processed on all machines in the same order. Also, the jobs can wait between two successive machines and the intermediate storage is unlimited. The completion time of job $j$, $j = 1, 2, \ldots, n$, on machine $i$, $i = 1, 2, \ldots, m$, is denoted by $C(i, j)$. First, for $\mathrm{F}m|r_j|C_{\max}$, let the makespan, the maximal completion time of the job on the final machine, of a dense schedule $S$ be $C_{\max}(S)$. And Assume the optimal makespan be $C_{\max}(S^*)$, where $S^*$ denotes the optimal schedule. Next, for $\mathrm{F}m||\Sigma C_j^2$, the objective value of an algorithm $H$ is denoted by $Z^H$, and the optimal solution is denoted by $Z^*$. The objective is to find a sequence of jobs, with the given processing times on each machine, to minimize the total quadratic completion times on the final machine, that is, min $\sum_{j=1}^{n} C^2(m, j)$.

## 3. Performance Analysis for Dense Schedule

To evaluate the performance of a heuristic algorithm, a classical way is to analyze its worst case performance ratio. But in an industrial scheduling environment, it is usual that thousands of jobs would be processed on one or more machines. As the appearance of worst case performance ratio is special for some small size problem, it is more suitable to introduce asymptotical performance ratio to describe the effect of a heuristic in practice when the problem size is large enough. In this section, the asymptotic optimality of dense schedule is shown by series of deductions.

As $\mathrm{F}m|r_j|C_{\max}$ is strongly NP hard, the usual way to estimate the optimal makespan is to calculate its lower bound. The classical lower bound (LB1), valid for this problem, is the bound equal to the maximum of machine loads on the first machine. The value of LB1 is

$$C_{\mathrm{LB1}} = \max_{1 \le j' \le n} \left\{ r_{j'} + \sum_{k=j'}^{n} p(k, 1) \right\}. \tag{3.1}$$

With the above preparation, it is easy to obtain the following theorem.

**Theorem 3.1.** *Let the processing times $p(i, j)$, $j = 1, 2, \ldots, n$, $i = 1, 2, \ldots, m$, be independent random variables having the same continuous distribution with bounded density $\phi(\cdot)$. For every $j$, $j = 1, 2, \ldots, n$, with probability one, one has that*

$$\lim_{n \to \infty} \frac{C_{\max}(S)}{n} = \lim_{n \to \infty} \frac{C_{\max}(S^*)}{n}. \tag{3.2}$$

*Proof.* For a dense schedule and its associated LB1, we have

$$
\begin{aligned}
C_{\max}(S) - C_{\mathrm{LB1}} &= \max_{1\le j\le l_1\le\cdots\le l_m\le n}\left\{ r_j + \sum_{k=j}^{l_1} p(k,1) + \sum_{k=l_1}^{l_2} p(k,2) + \cdots + \sum_{k=l_m}^{n} p(k,m) \right\} \\
&\quad - \max_{1\le j'\le n}\left\{ r_{j'} + \sum_{k=j'}^{n} p(k,1) \right\} \\
&\le \max_{1\le j\le l_1\le\cdots\le l_m\le n}\left\{ r_j + \sum_{k=j}^{l_1} p(k,1) + \sum_{k=l_1}^{l_2} p(k,2) + \cdots + \sum_{k=l_m}^{n} p(k,m) - r_j - \sum_{k=j}^{n} p(k,1) \right\} \\
&\le \max_{1\le j\le l_1\le\cdots\le l_{m-1}\le n}\left\{ \left( \sum_{k=j}^{l_1} p(k,1) + \cdots + \sum_{k=l_{m-1}}^{n} p(k,m) \right) - \sum_{k=j}^{n} p(k,1) \right\}.
\end{aligned}
\tag{3.3}
$$

The second inequality of (3.3) holds because of the following reasons. For a given schedule, it is obvious there is no extra idle time on the first machine except waiting for the arrivals. Therefore, in LB1 the release date $r_{j'}$ is the last one to generate the idle on that machine, and we have $r_j + p(j,1) + \cdots + p(j'-1,1) \le r_{j'}$.

For the last inequality of (3.3), the following cases are considered.

*Case 1* ($|n-j| \sim O(m)$). Let $n - j = \alpha m$, where $\alpha$ is a constant and $\lim_{n\to\infty}\alpha/n = 0$. Noting $n - l_1 = (\alpha+1)m - 1$, therefore, we have

$$
\begin{aligned}
&\lim_{n\to\infty}\frac{1}{n}\max_{1\le j\le l_1\le\cdots\le l_{m-1}\le n}\left\{ \left( \sum_{k=j}^{l_1} p(k,1) + \cdots + \sum_{k=l_{m-1}}^{n} p(k,m) \right) - \sum_{k=j}^{n} p(k,1) \right\} \\
&\quad = \lim_{n\to\infty}\frac{1}{n}\max_{1\le l_1\le\cdots\le l_{m-1}\le n}\left\{ \sum_{k=j}^{l_1} p(k,1) + \cdots + \sum_{k=l_{m-1}}^{n} p(k,m) \right\} - \lim_{n\to\infty}\frac{1}{n}\max_{1\le j\le n}\left\{ \sum_{k=j}^{n} p(k,1) \right\} \\
&\quad \le \lim_{n\to\infty}\frac{(\alpha+1)m-1}{n} p_{\max} - \lim_{n\to\infty}\frac{\alpha m}{n} p_{\min} = 0,
\end{aligned}
\tag{3.4}
$$

where $p_{\max}$ and $p_{\min}$ denote the maximum and the minimum processing time among all the processing times, respectively.

*Case 2* ($|n-j| \sim O(n)$). Let $n - j = \beta n$, where $0 < \beta < 1$ is a constant and $\lim_{n\to\infty}\beta/n = 0$. Noting $n - l_1 = \beta n + m - 1$, therefore, we have

$$
\begin{aligned}
&\lim_{n\to\infty}\frac{1}{n}\max_{1\le j\le l_1\le\cdots\le l_{m-1}\le n}\left\{ \left( \sum_{k=j}^{l_1} p(k,1) + \cdots + \sum_{k=l_{m-1}}^{n} p(k,m) \right) - \sum_{k=j}^{n} p(k,1) \right\} \\
&\quad \le \lim_{n\to\infty}\frac{1}{n}\max_{1\le l_1\le\cdots\le l_{m-1}\le n}\left\{ \sum_{k=j}^{l_1-1} p(k,1) + \sum_{k=l_1}^{l_2-1} p(k,2) \cdots + \sum_{k=l_{m-1}}^{n-1} p(k,m) \right\}
\end{aligned}
$$

$$+ \lim_{n\to\infty} \frac{m-1}{n} p_{\max} - \lim_{n\to\infty} \frac{1}{n} \max_{1\le j\le n} \left\{ \sum_{k=j}^{n} p(k,1) \right\}$$
$$= \lim_{n\to\infty} \beta E(p) - \lim_{n\to\infty} \beta E(p) + \lim_{n\to\infty} \frac{m-1}{n} p_{\max} = 0, \tag{3.5}$$

where $E(p)$ is the expected processing time of all the $n$ jobs. The last equality of (3.5) holds because that all processing times are i.i.d. random variables and satisfy the law of large numbers when the number of jobs trends to infinity.

With inequalities (3.4) and (3.5), and noting that

$$C_{\max}(S) - C_{\max}(S^*) \le C_{\max}(S) - C_{\text{LB1}}, \tag{3.6}$$

we have

$$0 \le \lim_{n\to\infty} \frac{C_{\max}(S) - C_{\max}(S^*)}{n} \le \lim_{n\to\infty} \frac{C_{\max}(S) - C_{\text{LB1}}}{n} = 0. \tag{3.7}$$

Rearranging inequality (3.7), we can get the result of the theorem. □

The above Theorem means that any dense permutation schedule for $\mathrm{F}m|r_j|C_{\max}$ approaches the optimal schedule when the scale of the problem is large enough.

## 4. Improvement Strategy for SPT Heuristic

In this section, the performance of classical SPT heuristic is modified by a local search scheme in which each job in the seed sequence is sequentially inserted in each possible different position of a generated sequence. However, as the asymptotic optimality of SPT sequence, the insertion movement of a job for objective descent may not be too far from its initial position, otherwise, it will enlarge the objective function value. Therefore, the searching and comparing for a seed sequence from front to back in the scheme will cost much redundant computation time. To save calculation time and produce high-quality solutions, two selected jobs are shifted from their current position, both ends of an initial sequence, and inserted in a different position forward and backward one by one, respectively. To guarantee the asymptotic optimality of the final solution, the initial sequence is generated by SPT heuristic. If the objective value is not improved after several insertions, we terminate the searching process for this seed sequence and then, choose the best sequence out of those sequences as the new seed sequence for the next two jobs to be inserted. Repeat the process of search, until all the jobs are tested.

Let $[x\colon (s)]$ denote the two jobs found in the $x$th and $n - x + 1$th position of sequence $\pi(s)$, and $[y\colon (s')]$ denote the two jobs found in the $y$th and $n - y + 1$th position of sequence $\pi(s')$. A formal description of the improvement strategy can be presented as follows.

*Improvement Strategy for SPT Heuristic*

*Step 1.* Generate the initial sequence $\pi(s)$ with SPT heuristic.

*Step 2.* Set $x = 1$ and $\pi(s) = \pi(s')$.

*Step 3.* Set $y = 1$.

*Step 4.* If $[x\colon (s)] \neq [y\colon (s')]$, then generate a sequence $\pi(y)$, which differs from $\pi(s')$ only by inserting two jobs $[x\colon (s)]$ to the $y$th and $n - y + 1$th position, and compute the quadratic completion time $C(y)$ of sequence $\pi(y)$; otherwise, set $y' = y$.

*Step 5.* Set $y := y + 1$. Return to Step 4 if $y \le n$; otherwise, go to Step 6.

*Step 6.* Determine the sequence $\pi(j)$ such that $C(j) = \min\{C(y) \mid y = 1, 2, \ldots, n, \text{ and } y' \neq y\}$. If $C(j) < C(s')$, set $\pi(s') = \pi(j)$.

*Step 7.* Set $x := x + 1$. return to Step 3 if $x \le n$; otherwise, go to Step 8.

*Step 8.* $\pi(s')$ is the final sequence. Stop.

To see the practical effectiveness of the improvement strategy, numerical simulations are conducted in the next section.

## 5. Computational Results

In this section, we designed a series of computational experiments to reveal the performances of the improvement strategy in different size problems. First, we compare the effectiveness of the improvement strategy with that of SPT heuristic; then, we report the ratios of the improvement strategy to the lower bound of $Fm\|\Sigma C_j^2$, LB2 (given by Koulamas and Kyparisis [7]):

$$\mathrm{LB2} = \frac{1}{m^2} \sum_{j=1}^{n} \left( \sum_{k=1}^{j} \sum_{i=1}^{m} p(k, i) \right)^2, \tag{5.1}$$

to show its performance variation when parameters vary. The ratios showed in Table 1 are the objective values of improvement strategy to that of SPT heuristic and its associated LB2 values, respectively. Three, five, and ten machines, and 100, 200, 500, and 1000 jobs are tested, respectively. The processing times were randomly generated from a discrete uniform distribution on [1, 10]. Five different random tests for each combination of the parameters mentioned above were performed, respectively, and the averages are shown in Table 1, where IS denote the improvement strategy.

From the data showed in Table 1, we can see that the ratios of IS/LB2 approach one as the number of jobs increases from 100 to 1000 with the fixed number of machines. For example, with five machines, the ratio of IS/LB2 drops from 1.5289 to 1.1609 when the number of jobs increases from 100 to 1000 (see Figure 1). This phenomenon indicates the asymptotic optimality of the improvement strategy. Contrarily, for the fixed number of jobs, ratios of IS/LB2 enlarges as the number of machines increases from 3 to 10. The cause may

**Table 1:** The experiment results.

| Machines | IS/LB2 | | | IS/SPT | | |
|---|---|---|---|---|---|---|
| | 3 | 5 | 10 | 3 | 5 | 10 |
| 100 jobs | 1.3956 | 1.5289 | 2.1135 | 0.9051 | 0.9121 | 0.9275 |
| 200 jobs | 1.1949 | 1.3235 | 1.5770 | 0.9205 | 0.9382 | 0.9564 |
| 500 jobs | 1.1610 | 1.2571 | 1.3224 | 0.9421 | 0.9656 | 0.9792 |
| 1000 jobs | 1.0941 | 1.1609 | 1.2186 | 0.9795 | 0.9831 | 0.9850 |

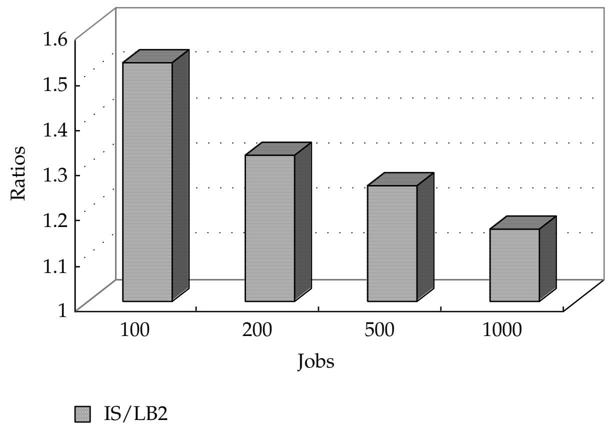


**Figure 1:** Ratios of IS/LB2 with $m = 5$.

be that the larger the number of machines is the larger the quantity of idle times is, which enlarges the gap between the value of objective and its lower bound.

From Table 1, we also find that the objective values of SPT heuristic and improvement strategy tend to be equal when the number of jobs is larger enough. An example can be see in Figure 2, with three machines, the ratio of IS/SPT enlarges from 0.9051 to 0.9795 when the number of jobs increases from 100 to 1000. A reasonable explanation is that the idle times between two adjacent jobs can be ignored and the associated objectives become nondifferentiated when the problem scale tends to infinity.

## 6. Conclusions

In this paper, we first showed that any dense permutation schedule is equal to the optimal schedule as the problem size tends to infinity for flow shop makespan problem with release dates. And then, an improvement strategy based on local search is presented to boost the performance of SPT heuristic for flow shop total quadratic completion time problem. Computational results show that the improvement strategy works well with the moderate scale problems.

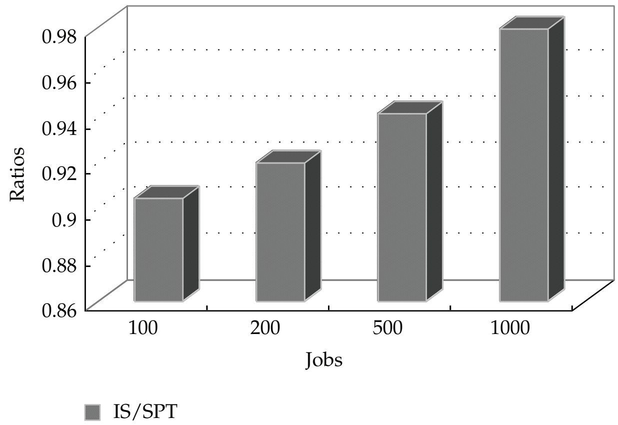


**Figure 2:** Ratios of IS/SPT with $m = 3$.


## Acknowledgments

First, the authors would express their appreciation to the anonymous referees and Professor Sotskov for their helpful comments which led to the improvement of this paper. This research is partly supported by National Natural Science Foundation of China (61104074, 60974091), Fundamental Research Funds for the Central Universities (N110417005, 100508001), China Postdoctoral Science Foundation (20100471462), Scientific Research Foundation for Doctor of Liaoning Province of China (20101032), 2011 Plan Project of Shenyang City (F11-264-1-63), Program for New Century Excellent Talents in University (NCET-08-0105), and the Scientific Research Foundation of State Education Ministry for the Returned Overseas Chinese Scholars.


## References


[1] B. Chen, C. N. Potts, and G. J. Woeginger, “A review of machine scheduling: complexity, algorithms and approximability,” in *Handbook of Combinatorial Optimization, Vol. 3*, D.-Z. Du and P. Pardalos, Eds., pp. 21–169, Kluwer Academic Publishers, Boston, Mass, USA, 1998.

[2] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, “Optimization and approximation in deterministic sequencing and scheduling: a survey,” *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[3] J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker, “Complexity of machine scheduling problems,” *Annals of Operations Research*, vol. 1, pp. 343–362, 1977.

[4] C. N. Potts, “Analysis of heuristics for two-machine flow-shop sequencing subject to release dates,” *Mathematics of Operations Research*, vol. 10, no. 4, pp. 576–584, 1985.

[5] L. A. Hall, “Approximability of flow shop scheduling,” *Mathematical Programming*, vol. 82, no. 1-2, pp. 175–190, 1998.

[6] T. C.E. Cheng and Z. Liu, “Parallel machine scheduling to minimize the sum of quadratic completion times,” *IIE Transactions*, vol. 36, no. 1, pp. 11–17, 2004.

[7] C. Koulamas and G. Kyparisis, “Algorithms with performance guarantees for flow shops with regular objective functions,” *IIE Transactions*, vol. 37, no. 12, pp. 1107–1111, 2005.

[8] N. M. Matsveichuk, Y. N. Sotskov, N. G. Egorova, and T.-C. Lai, “Schedule execution for two-machine flow-shop with interval processing times,” *Mathematical and Computer Modelling*, vol. 49, no. 5-6, pp. 991–1011, 2009.

[9] C. T. Ng, N. M. Matsveichuk, Y. N. Sotskov, and T. C. E. Cheng, “Two-machine flow-shop minimum-length scheduling with interval processing times,” *Asia-Pacific Journal of Operational Research*, vol. 26, no. 6, pp. 715–734, 2009.

[10] D. Bai and L. Tang, “Two asymptotically optimal lower bounds for flow shop problem,” *Chinese Journal of Electronics*, vol. 18, no. 4, pp. 625–629, 2009.

[11] D. Bai and L. Tang, “New heuristics for flow shop problem to minimize makespan,” *Journal of the Operational Research Society*, vol. 61, no. 6, pp. 1032–1040, 2010.

[12] I. Bárány and T. Fiala, “Nearly optimum solution of multimachine scheduling problems,” *Szigma*, vol. 15, no. 3, pp. 177–191, 1982 (Hungarian).

[13] B. Chen and W. Yu, “How good is a dense shop schedule?” *Acta Mathematicae Applicatae Sinica*, vol. 17, no. 1, pp. 121–128, 2001.