How Good Is A Dense Shop Schedule?

Bo Chen[*] Wenci Yu[†]

July 1998

Abstract

In this paper, we study a class of simple and easy-to-construct shop schedules, known as dense schedules. We present tight bounds on the maximum deviation in makespan of dense flow-shop and job-shop schedules from their optimal ones. For dense open-shop schedules, we do the same for the special case of four machines and thus add a stronger supporting case for proving a standing conjecture.

Keywords: shop scheduling, dense schedules, worst-case performance

1 Introduction

In the shop scheduling model we are given $m \geq 2$ machines and n jobs. Each job consists of a number of operations, each to be processed on a specified machine for a specific amount of time. Each machine processes no more than one job at a time, and each job is processed on at most one machine at a time. No preemption is allowed in processing any operation, i.e., once started, the processing of any operation must not be interrupted before it is completed.

Traditionally, in scheduling theory three basic shop scheduling models are considered: the job-shop, flow-shop and open-shop. In the *job-shop*, each job consists of a chain of a number of operations, each of which should be processed by a specified machine. The *flow-shop* is a special case of the job-shop where the chain of each job has m operations, each of which is assigned to exactly one machine, and the machine order as specified by the chain of each job is the same. The *open-shop* differs from the flow-shop in that, for

---

[*] Warwick Business School, University of Warwick, Coventry CV4 7AL, England, U.K.

[†] Institute of Applied Mathematics, East China University of Science and Technology, Shanghai 200237, P.R. China.

each job, the order of its operations is not specified in advance and is itself part of the decision to make for the scheduler, different jobs being allowed to get different orders.

For each of the three models, the most well-known problem is to minimize the makespan, i.e., the time when the last job is completed. All these three problems are strongly NP-hard (see, e.g., [4]). Therefore, it is an interesting research goal to develop quality polynomial approximation algorithms. Usually, the quality of a polynomial approximation algorithm is measured by its *worst-case performance ratio* $\sup C_{\max}(S)/C^*_{\max}$, where $C_{\max}(S)$ is the makespan of a schedule S found by the algorithm in question, $C^*_{\max}$ is the corresponding minimum possible makespan, and the supremum is taken over all problem instances.

For the flow-shop problem, $O(mn + n\log n)$ time approximation algorithms are presented in [5, 6] with worst-case performance ratios no more than $\lceil m/2 \rceil$. Similar approximation result for the open-shop problem is obtained in [6]. Randomized polynomial approximation algorithms for the job-shop problem are constructed in [8] with worst-case performance ratios of $O(\log^2(m\mu)/\log\log(m\mu))$, where μ is the maximum number of operations per job. The same performance guarantee is achieved by a deterministic polynomial approximation algorithm in [7].

Recently, it has been discovered that finding a guaranteed good approximate solution for any shop scheduling problem is as difficult as finding an optimal one. It is shown in [10] that, unless P=NP, there is no polynomial approximation algorithm for any of these shop scheduling problems with a worst-case performance ratio less than $5/4$.

Nevertheless, there is an interesting simple class of shop schedules that can be constructed by any straightforward greedy algorithm. A feasible shop schedule is called *dense* when any machine is idle if and only if there is no job that can be processed at that time on that machine. It is observed (see [2]) that the makespan of any dense open-shop schedule is at most twice of the optimal one. It is conjectured that, for every $m \geq 2$, the makespan of a dense open-shop schedule is at most $2 - 1/m$ times the optimal makespan, which is proved in [3] for $m \leq 3$. It is shown in [3, 9] that the bound of $2 - 1/m$ is actually achieved.

Besides a guaranteed relatively good quality, a dense shop schedule can be constructed very easily by any greedy algorithm. More interestingly, such a construction can be carried out under an *on-line* and *non-clairvoyant* environment, i.e., without any knowledge of processing requirement of any unfinished operation and without any knowledge of future jobs.

In this paper, we prove that the conjecture that any dense open-shop schedule has a makespan at most $2 - 1/m$ times the optimum is true also for

$m = 4$. Furthermore we establish that any dense schedule for job-shop and flow-shop has a makespan at most m times the optimum and the bound of m is best possible.

2 Notation and preliminaries

Let us formally describe the three shop scheduling models. Let $\{M_1, \ldots, M_m\}$ be the set of machines and $\{J_1, \ldots, J_n\}$ be the set of jobs. In the open-shop, each job J_j consists of a *set* $\{O_{1,j}, \ldots, O_{m,j}\}$ of operations, while in the flow-shop, each job J_j consists of a *chain* $(O_{1,j}, \ldots, O_{m,j})$ of operations. In both shops, operation $O_{i,j}$ has to be processed on machine M_i for $p_{i,j}$ time units. In the job-shop, each job J_j is a chain $(O_{1,j}, \ldots, O_{m_j,j})$ of operations, and $O_{i,j}$ has to be processed on machine $M_{\mu_{i,j}}$ for $p_{i,j}$ time units.

The *workload* P_i of machine M_i is the total processing time of all operations assigned to the machine, i.e., $P_i = \sum_{j=1}^{n} p_{i,j}$ in the case of open-shop or flow-shop and

$$P_i = \sum_{j=1}^{n} \sum_{\substack{k:\, 1 \le k \le m_j \\ \mu_{k,j} = i}} p_{k,j}$$

in the case of job-shop. The *length* p_j of job J_j is the total processing time of all the operations of the job, i.e., $p_j = \sum_{i=1}^{m} p_{i,j}$ in the case of open-shop or flow-shop, and $p_j = \sum_{i=1}^{m_j} p_{i,j}$ in the case of job-shop. Define

$$L = \max\{\max_{1 \le i \le m} P_i,\ \max_{1 \le j \le n} p_j\}.$$

Apparently, L is a lower bound on the optimal makespan.

Given any dense schedule, let $B_{i,j}$ and $C_{i,j}$ be the start and completion time of operation $O_{i,j}$, respectively. Hence $C_{i,j} = B_{i,j} + p_{i,j}$. A (time) *interval* is a set of reals of the form $[a, b) = \{t :\ a \le t < b\}$, where $b > a \ge 0$. Hence, we said that $[a, b)$ is an *idle interval* up to time $t \ge b$ on a machine if on this machine there is no processing since a job finishes at time a until time b if $b = t$ or until another job starts at time b if $b < t$. A machine is said to be idle *at* time t if t belongs to an idle interval on the machine. Denote by $I(O_{i,j})$ the whole processing interval of operation $O_{i,j}$, i.e., $I(O_{i,j}) = [B_{i,j}, C_{i,j})$. Let $C_{\max}(S)$ and $C^*_{\max}$ denote the makespan of a schedule S and the optimal makespan, respectively. For any two sets of intervals, $\mathcal{R}' = \{I'\}$ and $\mathcal{R}'' = \{I''\}$, define $\mathcal{R}' \tilde{\cap} \mathcal{R}''$ as the set of intervals $\{I' \cap I'' :\ I' \in \mathcal{R}', I'' \in \mathcal{R}''\}$. Let $\sigma(\mathcal{R})$ denote the total length of intervals in $\mathcal{R}$ and, in particular, let it be denoted by τ_i if $\mathcal{R}$ is the set of idle intervals on machine M_i up to time $C_{\max}(S)$ for $i = 1, \ldots, m$.

3 On open-shops

Since L is a lower bound of the optimal makespan, it is evident that bounding the machine idle times is crucial in analyzing the worst-case deviation in makespan. We first establish the following four lemmas to help this bounding procedure.

Lemma 1 *Suppose that* $C_{\max}(S) > (2-1/m)L$ *for dense open-shop schedule* S*, and that machine* M_m *terminates the schedule. Then there is at least one idle interval on* M_m*. Let* G *be the set of jobs that are processed on* M_m *after the last idle interval on the machine. Then* $|G| \leq m-1$ *and* $p_{m,j} < p_j/m$ *for any* $J_j \in G$*.*

Proof: If there is no idle time on M_m, then S is optimal, since $C_{\max}(S) = P_m$. Since all the jobs in G are processed *simultaneously* on other machines right before they are started on M_m (otherwise, at least one of them could have been processed earlier), we have $|G| \leq m-1$. Since all the idle intervals on M_m are contained in $\cup_{i=1}^{m-1} I(O_{i,j})$ for any $J_j \in G$, that $p_{m,j} \geq p_j/m$ would imply that the total idle time on M_m would be no more than $(m-1)p_j/m$ and hence $C_{\max}(S) \leq (2-1/m)L$. ■

Lemma 2 *In addition to the assumptions in Lemma 1, we further assume that job* J_1 *finishes last on machine* M_m *and it is processed on machines* $M_1, \ldots, M_m$ *in this order. If job* J_1 *is* continuously *processed on machines* M_k *up to* M_m*, where* k *is taken as small as possible, then the following two statements are true:*

(a) *Among* $\{I(O_{i,1}), i = 1, \ldots, k-1\}$*, there is at least one interval of which machines* $M_k, \ldots, M_m$ *have a common idle subinterval.*

(b) $k \geq \lceil (m+3)/2 \rceil$.

Proof: Let τ_i' be the total idle time on machine M_i up to time $B_{k,1}$. Then for any $k \leq i \leq m$, we have

$$C_{\max}(S) \leq P_i + \tau_i' + \sum_{l=k}^{m} p_{l,1} - p_{i,1}. \tag{1}$$

Since $\tau_i' \leq \sum_{i=1}^{k-1} p_{i,1}$, violation of (a) would imply that

$$\sum_{i=k}^{m} \tau_i' \leq (m-k) \sum_{i=1}^{k-1} p_{i,1},$$

which, together with the $m-k+1$ inequalities in (1), contradicts our original assumption for the lower bound of $C_{\max}(S)$. Furthermore, since all machines

$M_k, \ldots, M_m$ are busy during $[C_{k-1,1}, B_{k,1})$, they are processing $(m-k+1)$ different jobs right before time $B_{k,1}$. Hence statement (a) implies that during a period within some interval $I(O_{i,1})$ $(1 \leq i \leq k-1)$ the $(m-k+1)$ different jobs are being processed on machines M_1 up to M_{k-1} but excluding M_i, which implies that $(m-k+1) \leq k-2$ and hence (b). ■

We now introduce the concept of coverage for our future analysis of machine idle times. Let $\mathcal{R}$ and $\mathcal{Q}$ be two sets of intervals. For any $t \geq 0$, let $\mathcal{R}(t)$ and $\mathcal{Q}(t)$ be the number of intervals in $\mathcal{R}$ and $\mathcal{Q}$, respectively, that contain t. If $\mathcal{R}(t) \leq \mathcal{Q}(t)$ for any $t \geq 0$, then $\mathcal{R}$ is said to be *covered* by $\mathcal{Q}$. It is easy to see that the following lemma holds.

Lemma 3 *Let $\mathcal{R}$, $\mathcal{R}'$, $\mathcal{R}''$, $\mathcal{Q}$, $\mathcal{Q}'$ and $\mathcal{Q}'$ be sets of intervals. If $\mathcal{R}'$ is covered by $\mathcal{Q}'$, $\mathcal{R}''$ is covered by $\mathcal{Q}''$, and $\mathcal{R}'\tilde{\cap}\mathcal{R}'' = \emptyset$, then set $\{\mathcal{R}', \mathcal{R}''\}$ is covered by set $\{\mathcal{Q}', \mathcal{Q}''\}$. If $\mathcal{R}$ is covered by $\mathcal{Q}$, then their total interval lengths satisfy $\sigma(\mathcal{R}) \leq \sigma(\mathcal{Q})$.*

The following lemma shows how an upper bound on the makespan deviation of a shop schedule can be derived from an estimate of the machine idle times in the schedule.

Lemma 4 *Let S be any shop schedule. If there is a vector $(w_1, \ldots, w_m)$, where $w_i \geq 0$ for all $i = 1, \ldots, m$, such that $\sum_{i=1}^{m} w_i \tau_i \leq (w-1)L$, then it holds that $C_{\max}(S) \leq (2 - w^{-1})C^*_{\max}$, where $w = \sum_{i=1}^{m} w_i$.*

Proof: For each $1 \leq i \leq m$, we have $C_{\max}(S) = P_i + \tau_i \leq L + \tau_i$. Taking a weighted summation with respect to $(w_1, \ldots, w_m)$, we reach the conclusion. ■

The remainder of this section is devoted to considering the special case of four machines. We will prove the following statement.

Theorem 5 *The makespan of any dense schedule for the four-machine open-shop is at most $7/4$ times that of an optimal schedule.*

Proof of Theorem 5

Consider any dense schedule for the four-machine open-shop. Without loss of generality, we assume that job J_1 terminates the schedule and its last operation is processed on machine M_4. As before let G be the set of jobs that are processed on M_4 after the last idle interval. Then statement (b) of

Lemma 2 implies that we can assume $|G| \geq 2$, and Lemma 1 allows us to be only concerned with that $|G| \leq 3$ and that

$$p_{4,j} < L/4, \text{ for all } J_j \in G.$$

First let us consider the situation in which G consists of three jobs. Assume without loss of generality that the other two jobs in the group are job J_2 as second last and job J_3 as the third last. The following two facts can be easily observed: (i) While machine M_4 is idle, all the other three machines are busy with processing jobs J_1, J_2 and J_3. Assume that jobs J_1, J_2 and J_3 are being processed on machines M_i, M_k and M_l, respectively, right before time $B_{4,3}$ ($i, k, l = 1, 2, 3$ and are different from each other). (ii) Any two of machines M_1, M_2 and M_3 cannot be idle *simultaneously* before time $B_{4,3}$, otherwise one of the operations $O_{i,1}$, $O_{k,2}$ and $O_{l,3}$ could be processed earlier. Therefore, we conclude that all the idle intervals on any machine before time $B_{4,3}$ do not intersect those on the other machines. Namely, if we let $\mathcal{R}^i$ be the set of idle time intervals on M_i up to time $B_{4,3}$ for $i = 1, \ldots, 4$, then $\mathcal{R}^i \cap \mathcal{R}^k = \emptyset$ when $i \neq k$ and $1 \leq i, k \leq 4$. Let $\mathcal{R}^5$ and $\mathcal{R}^6$ be two copies of $\mathcal{R}^4$, $\mathcal{R} = \{\mathcal{R}^1, \ldots, \mathcal{R}^6\}$ and $\mathcal{Q} = \{I(O_{i,j}) : i, j = 1, 2, 3\}$. Since $\mathcal{R}^i$ is covered by $\mathcal{Q}$ for all $i = 1, 2, 3$, and set $\{\mathcal{R}^4, \mathcal{R}^5, \mathcal{R}^6\}$ is also covered by $\mathcal{Q}$ due to the aforementioned fact (i), we conclude, according to Lemma 3, that $\mathcal{R}$ is covered by $\mathcal{Q}$ and

$$\sum_{i=1}^{3} \sigma(\mathcal{R}^i) + 3\sigma(\mathcal{R}^4) \leq \sum_{i=1}^{3} \sum_{j=1}^{3} p_{i,j},$$

which implies

$$\sum_{i=1}^{3} \tau_i/3 + 3\tau_4 \leq \sum_{i=1}^{3} \sum_{j=1}^{3} p_{i,j} + \sum_{j=1}^{3} p_{4,j} = \sum_{j=1}^{3} p_j \leq 3L.$$

Therefore, in this case we are done according to Lemma 4.

Now consider the other situation in which G consists of two jobs. Without loss of generality, we assume that $G = \{J_1, J_2\}$, which implies that the last idle interval on machine M_4 finishes at $B_{4,2}$. Let $\mathcal{R}^i$ be the set of the idle intervals on M_i up to time $B_{4,2}$ for $i = 1, \ldots, 4$. Without loss of generality, let jobs J_2 and J_1 be processed right before time $B_{4,2}$ on machines M_2 and M_3, respectively. It is immediately seen that $\mathcal{R}^2 \tilde{\cap} \mathcal{R}^3 = \emptyset$, since otherwise either operation $O_{3,1}$ or $O_{2,2}$ could be processed earlier. Consider the following three conditions:

- Condition 1: Either $R^2 = \emptyset$ or $R^3 = \emptyset$.

- Condition 2: $\{\mathcal{R}^2, \mathcal{R}^3\} \tilde{\cap} \mathcal{R}^4 = \emptyset$.
- Condition 3: $\{\mathcal{R}^2, \mathcal{R}^3\} \tilde{\cap} \mathcal{R}^4 \neq \emptyset$, and $\mathcal{R}^1 \tilde{\cap} \{\mathcal{R}^2, \mathcal{R}^3\} = \emptyset$.

If Condition 1 is satisfied, then Theorem 5 is proved by the fact that

$$C_{\max}(S) \leq \max_{2 \leq i \leq 3} P_i + (p_{4,1} + p_{4,2}) \leq L + L/4 + L/4 < 7L/4.$$

Let $\mathcal{Q} = \{I(O_{i,j}) : i = 1, 2, 3; j = 1, 2\}$. Under Condition 2, let $\mathcal{R}^5$ be a copy of $\mathcal{R}^4$. Since any of $\mathcal{R}^2$, $\mathcal{R}^3$ and $\{\mathcal{R}^4, \mathcal{R}^5\}$ is covered by $\mathcal{Q}$, according to Lemma 3, we conclude that $\{\mathcal{R}^2, \mathcal{R}^3, \mathcal{R}^4, \mathcal{R}^5\}$ is covered by $\mathcal{Q}$ and

$$\begin{aligned} \tau_2 + \tau_3 + 2\tau_4 &\leq \sigma(\mathcal{R}^2) + \sigma(\mathcal{R}^3) + 2\sigma(\mathcal{R}^4) + 2(p_{4,2} + p_{4,1}) \\ &\leq \sum_{i=1}^{3} \sum_{j=1}^{2} p_{i,j} + 2(p_{4,2} + p_{4,1}) \leq p_1 + p_2 + (L/4 + L/4) < 3L, \end{aligned}$$

which, together with Lemma 4, implies that we are done.

Finally, suppose Condition 3 holds. Let $[a, b)$ be the last interval of $\{\mathcal{R}^2, \mathcal{R}^3\} \tilde{\cap} \mathcal{R}^4$. (Note that this is well defined.) Then there is no idle time on both machines M_2 and M_3 from time b to $B_{4,2}$ and the only jobs to be processed after time b on M_2 and M_3 are jobs J_1 and J_2. Therefore, we conclude that $\{\mathcal{R}^2, \mathcal{R}^3\} \tilde{\cap} \{[b, B_{4,2})\} = \emptyset$ and that $\{\mathcal{R}^1, \mathcal{R}^4\} \tilde{\cap} \{[b, B_{4,2})\}$ is covered by $\mathcal{Q}$. Since machine M_1 is processing job J_1 or J_2 right before time b, it is implied by $\mathcal{R}^1 \tilde{\cap} \{\mathcal{R}^2, \mathcal{R}^3\} = \emptyset$ that $\{\mathcal{R}^1, \mathcal{R}^2, \mathcal{R}^3, \mathcal{R}^4\} \tilde{\cap} \{[0, b)\}$ is covered by $\mathcal{Q}$ too. Hence, $\{\mathcal{R}^1, \mathcal{R}^2, \mathcal{R}^3, \mathcal{R}^4\}$ is covered by $\mathcal{Q}$ and, according to Lemma 3, we obtain

$$\begin{aligned} \sum_{i=1}^{4} \tau_i &\leq \sum_{i=1}^{3} \sum_{j=1}^{2} p_{i,j} + 3(p_{4,2} + p_{4,1}) \\ &= (p_1 + p_2) + 2(p_{4,2} + p_{4,1}) \leq 3L, \end{aligned}$$

which, together with Lemma 4, again shows that we are done.

We complete our theorem proof by showing that violation of both Conditions 2 and 3, i.e.,

$$\{\mathcal{R}^2, \mathcal{R}^3\} \tilde{\cap} \mathcal{R}^4 \neq \emptyset \text{ and } \mathcal{R}^1 \tilde{\cap} \{\mathcal{R}^2, \mathcal{R}^3\} \neq \emptyset, \tag{2}$$

will result in a satisfaction of Condition 1. Let $[a, b)$ be defined as above, and let t be any time that belongs to an interval of the non-empty set in (2). Then, as we have seen, either job J_2 or J_1 is being processed on machine M_1 right before time b.

Consider first the former case. While M_1 is idle at time $t < b$, it is (among other things) waiting for the completion of job J_2 on some machine.

However, such a processing of J_2 has to be on machine M_3 as can be easily observed, which together with the definition of t implies that M_2 is idle at the same time t. Note that job J_1 must have already been completed on both M_1 and M_2 before time t, i.e., $C_{1,1} < t$ and $C_{2,1} < t$, otherwise it could be processed earlier at t on either machine. This, together with the fact that J_1 starts earlier on M_3 than on M_4, implies that there is no idle time on machine M_3 between jobs J_2 and J_1, otherwise J_1 could be processed earlier since $C_{3,2} \geq t$. Therefore, we conclude that $\mathcal{R}^3 = \emptyset$ from the fact that M_3 as the first machine for processing job J_2 leaves no idle time before starting J_2.

Proof for the latter case, where job J_1 is being processed on machine M_1 right before time b, is completely parallel to that for the former case, and one will be led to a satisfaction of Condition 1 with $\mathcal{R}^2 = \emptyset$.

4 On flow-shop and job-shop

Now let us come back to considering the general case of m machines. In the cases of flow-shop and job-shop, we are able to fully understand the worst-case makespan deviation of dense schedules from the optimum. Let us first provide an upper bound on such a deviation for the case of job-shop.

Lemma 6 *For any dense job-shop schedule S, it holds that*

$$C_{\max}(S)/C^*_{\max} \leq m.$$

Proof: Without loss of generality, we assume that machine M_m terminates the schedule. Note that M_m is idle only if at least one of machines $M_1, \ldots, M_{m-1}$ is busy. Therefore, the set of idle intervals on M_m satisfies that

$$\mathcal{R}_m \subseteq \bigcup_{j=1}^{n} \bigcup_{\substack{i=1 \\ \mu_{i,j} \neq m}}^{m_j} I(O_{i,j}),$$

which implies that $\tau_m \leq \sum_{i=1}^{m-1} P_i \leq (m-1)L$, and thus that $C_{\max}(S) = P_m + \tau_m \leq mL$. ■

The above bound of m is actually achievable even for the flow-shop, a special case of job-shop, and even by a *permutation* dense schedule for the special job-shop, where a permutation flow-shop schedule is one in which the m machines process the jobs in the same sequence given by a permutation.

Lemma 7 *For any given positive value ϵ, there exists a flow-shop instance and a dense permutation schedule S for the instance, such that*

$$C_{\max}(S)/C^*_{\max} > m - \epsilon.$$

Proof: Let us start with the simple instance $\mathcal{I}$ in which the machine order for each job is $(M_1, \ldots, M_m)$ and the processing time of every operation is one unit. A dense permutation schedule S_ϕ, which is based on permutation $\phi = (1, \ldots, n)$ and is itself optimal, is to start operation $O_{i,j}$ at time $i + j - 2$ $(i = 1, \ldots, m;\ j = 1, \ldots, n)$. Now we construct a new instance $\mathcal{I}'$ by simply increasing the processing times of the m diagonal operations $O_{i,i}$ from 1 to $p > 1$ for $i = 1, \ldots, m$. Consider the permutation schedule S, which is obtained by *pushing* the operations to the right as little as necessary to keep feasibility of the schedule, while maintaining the job sequence $(J_1, \ldots, J_n)$ on each machine. Namely, the starting times of the operations in S are as follows:

$$B_{i,j} = \begin{cases} ip + j - 2 & \text{if } i < j, \\ (i-1)(p+1) & \text{if } i = j, \\ jp + i - 2 & \text{if } i > j. \end{cases}$$

Apparently schedule S remains dense and permutational, and its makespan is $C_{\max}(S) = mp + m - 1$.

On the other hand, an optimal schedule T, which is also permutational, for the instance $\mathcal{I}'$ can be obtained in the same way from a dense permutation schedule for instance $\mathcal{I}$ as S is obtained except that it is based on another permutation $\psi = (\psi(1), \ldots, \psi(n)) = (n, \ldots, 1)$. Namely, the starting times of the operations in schedule T are as follows:

$$B_{\psi(i),\psi(j)} = \begin{cases} i + j - 2 & \text{if } i + j \le m + 1, \\ p + i + j - 3 & \text{otherwise.} \end{cases}$$

It is easy to calculate that $C_{\max}(T) = p + 2m - 2$. Therefore,

$$C_{\max}(S)/C^*_{\max} = C_{\max}(S)/C_{\max}(T) = (mp + m - 1)/(p + 2m - 2),$$

which is greater than $m - \epsilon$ when p is sufficiently large. ■

As an immediate consequence, we conclude in the following theorem that a dense schedule for a flow-shop or job-shop is an m factor away in makespan from the optimal.

Theorem 8 *For any dense schedule S for a flow-shop or job-shop, it holds that*

$$C_{\max}(S)/C^*_{\max} \le m,$$

and the bound m is tight.

Proof: Since flow-shop is a special case of job-shop, the first part of the theorem is implied by Lemma 6, and the second part is implied by Lemma 7. ■

5 Closing remarks

In this paper, we have analyzed the maximum deviation of the makespan of any dense shop schedule from the optimal one. The bounds we have proved are all tight. Unfortunately, we are still not able to prove, for *general* m machines, the conjecture that any dense open-shop has a makespan that is at most $2 - 1/m$ times the optimal one. However, such a proof now seems not out of sight.

Acknowledgement

Support for the research by the first author is provided by the Management Research Fellowship of the ESRC (Economic & Social Research Council) of U.K. Main part of the second author's work was done while he was a visiting professor at the Department of Mathematics of the City University of Hong Kong, to which the author is very grateful for its support. The author's research is also partially supported by the National Natural Science Foundation of China.

References

[1] V.A. Aksjonov, A polynomial-time algorithm for an approximate solution of a scheduling problem (in Russian), *Upravlyaemye Sistemy* 28 (1988), 8–11.

[2] I. Bárány and T. Fiala, Nearly optimum solution of multimachine scheduling problems (in Hungarian), *Szigma Mathematika Közgazdasági Folyóirat* 15 (1982), 177–191.

[3] B. Chen and V.A. Strusevich. Approximation algorithms for three machine open shop scheduling. *ORSA Journal on Computing* 5 (1993), 321–326.

[4] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, Sequencing and scheduling: algorithms and complexity, in S. Graves, A.H.G. Rinnooy Kan and P. Zipkin (eds.) *Handbooks in Operations Research and Management Science, Volume 4, Logistics of Production and Inventory*, North Holland, Amsterdam, 1993, 445-522.

[5] E. Nowicki and C. Smutnicki, Worst-case analysis of an approximation algorithm for flow-shop scheduling, *Operations Research Letters* 8 (1989), 171-177.

[6] H. Röck and G. Schmidt, Machine aggregation heuristics in shop-scheduling, *Methods of Operations Research* 45 (1983), 303–314.

[7] J.P. Schmidt, A. Siegel and A. Srinivasan, Chernoff-Hoeffding bounds for applications with limited independence, *SIAM Journal on Discrete Mathematics* 8 (1995), 223–250.

[8] D.B. Shmoys, C. Stein and J. Wein, Improved approximation algorithms for shop scheduling problems, *SIAM Journal on Computing* 23 (1994), 617–632.

[9] J.M. Wein, *Algorithms for Scheduling and Network Problems*, Ph.D. Thesis, MIT, Cambridge, MA, 1991.

[10] D.P. Williamson, L.A. Hall, J.A. Hoogeveen, C.A.J. Hurkens, J.K. Lenstra, S.V. Sevastianov and D.B. Shmoys, Short shop schedules, *Operations Research* 45 (1997), 288–294.