Improved Bounds for Flow Shop Scheduling

Monaldo Mastrolilli and Ola Svensson

IDSIA - Switzerland. {monaldo,ola}@idsia.ch

Abstract. We resolve an open question raised by Feige & Scheideler by showing that the best known approximation algorithm for flow shops is essentially tight with respect to the used lower bound on the optimal makespan. We also obtain a nearly tight hardness result for the general version of flow shops, where jobs are not required to be processed on each machine.

Similar results hold true when the objective is to minimize the sum of completion times.

1 Introduction

In the *flow shop scheduling problem* we have a set of n jobs that must be processed on a given set of m machines that are located in a fixed order. Each job j consists of a sequence of m operations, where the i-th operation must be processed during $p_{ij} \in \mathbb{Z}^+$ time units without interruption on the i-th machine. A feasible schedule is one in which each operation is scheduled only after all operations preceding it in its job have been completed, and each machine processes at most one operation at a time. A natural generalization of the flow shop problem is to not require jobs to be processed on all machines, i.e., a job still requests the machines in compliance with their fixed order but may skip some of them. We will refer to this more general version as *generalized flow shops* or *flow shops with jumps*. Generalized flow shop (and thus flow shop) scheduling is a special case of the *acyclic job shop* scheduling, which only requires that within each job all operations are performed on different machines, which in turn is a special case of the general *job shop* scheduling, where each job may have many operations on a single machine.

For any given schedule, let C_j be the completion time of the last operation of job j. We consider the natural and typically considered objectives of minimizing the *makespan*, $C_{\max} = \max_j C_j$, and the *sum of weighted completion times*, $\sum w_j C_j$, where w_j are positive integers. The goal is to find a feasible schedule which minimizes the considered objective function. In the notation of Graham et al. [6] the flow shop scheduling problem is denoted as $F||\gamma$, where γ denotes the objective function to be minimized. We will sometimes abbreviate the generalized flow shop problem by $F|jumps|\gamma$.

1.1 Literature Review

Flow shops have long been identified as having a number of important practical applications and have been widely studied since the late 50's (the reader is referred to the survey papers of Lawler et al. [9] and of Chen, Potts & Woeginger [2]). To find a schedule that minimizes the makespan, or one that minimizes the sum of completion times, was proved to be strongly NP-hard in the 70's, even for severely restricted instances (see e.g. [2]).

From then, many approximation methods have been proposed. Since the quality of an approximation algorithm is measured by comparing the returned solution value with a polynomial time computable lower bound on the optimal value, the goodness of the latter is very important. For a given instance, let C^*_{max} denote the minimum makespan taken over all possible feasible schedules. If D denotes the length of the longest job (the dilation), and C denotes the time units requested by all jobs on the most loaded machine (the congestion), then $lb = max[C, D]$ is a known trivial lower bound on C^*_{max}. To the best of our knowledge, no significant stronger lower bound is known on C^*_{max}, and all the proposed approximation algorithms for flow shops (but also for the more general job shop, acyclic job shop and the more constrained case of permutation flow shops) have been analyzed with respect to this lower bound (see, e.g., [10, 11, 4, 17, 5, 13]).

Even though the trivial lower bound might seem weak a surprising result by Leighton, Maggs & Rao [10] says that for acyclic job shops, if all operations are of unit length, then $C^*_{max} = \Theta(lb)$. If we allow operations of any length, then Feige & Scheideler [4] showed that $C^*_{max} = O(lb \cdot \log lb \log\log lb)$. They also showed their analysis to be nearly tight by providing acyclic job shop instances with $C^*_{max} = \Omega(lb \cdot \log lb / \log\log lb)$. The proofs of the upper bounds in [10, 4] are nonconstructive and make repeated use of (a general version) of Lovasz local lemma. Algorithmic versions appeared in [1, 3]. Recently, the authors showed that the best known approximation algorithm for acyclic job shops is basically tight [12]. More specifically, it was shown that for every $\epsilon > 0$, the (acyclic) job shop problem cannot be approximated within ratio $O(\log^{1-\epsilon} lb)$, unless NP has quasi-polynomial Las-Vegas algorithms.

In contrast to acyclic job shops, the strength of the lower bound lb for flow shop scheduling is not well understood, and tight results are only known for some variants. A notable example is given by the *permutation flow shop* problem, that is a flow shop problem with the additional constraint that each machine processes all the jobs in the same order. Potts, Shmoys & Williamson [14] gave a family of permutation flow shop instances with $C^*_{max} = \Omega(lb \cdot \sqrt{\min[m, n]})$. This lower bound was recently showed to be tight, by Nagarajan & Sviridenko [13], who gave an approximation algorithm that returns a permutation schedule with makespan $O(lb \cdot \sqrt{\min[m, n]})$.

Feige & Scheideler's upper bound for acyclic jobs [4] is also the best upper bound for the special case of flow shops. As flow shops have more structure than acyclic job shops and no flow shop instances with $C^*_{max} = \omega(lb)$ were known, one could hope for a significant better upper bound for flow shops. The existence of such a bound was raised as an open question in [4]. Unfortunately our recent inapproximability results for acyclic job shops do not apply to flow shops, since in [12] our construction builds upon the lower bound construction for acyclic job shop, which does not seem to generalize to flow shop [4]. The only known inapproximability result is due to Williamson et al. [18], and states that when the number of machines and jobs are part of the input, it is NP-hard to approximate $F||C_{\max}$ with unit time operations, and at most three operations per job, within a ratio better than 5/4. It is a long standing open problem if the above algorithms $F||C_{\max}$, are tight or even nearly tight (see, e.g. "Open problem 6 " in [16]).

A similar situation holds for the objective $\sum w_j C_j$. Queyranne & Sviridenko [15] showed that an approximation algorithm for the above mentioned problems that produces a schedule with makespan a factor $O(\rho)$ away from the lower bound lb can be used to obtain a $O(\rho)$-approximation algorithms for other objectives, including the sum of weighted completion times. The only known inapproximability result is by Hoogeveen, Schuurman & Woeginger [7], who showed that $F||\sum C_j$ is NP-hard to approximate within a ratio better than $1+\epsilon$ for some small $\epsilon > 0$.

1.2 Our Results

In this paper, we show that the best known upper bound [4] is essentially the best possible, by proving the existence of instances of flow shop scheduling for which the shortest feasible schedule is of length $\Omega(lb \cdot \log lb / \log\log lb)$. This resolves (negatively) the aforementioned open question by Feige & Scheideler [4].

Theorem 1. *There are flow shop instances for which any schedule has makespan* $\Omega(lb \cdot \log lb / \log\log lb)$.

If we do not require a job to be processed on *all* machines, i.e. generalized flow shops, we prove that it is hard to improve the approximation guarantee. Theorem 2 shows that generalized flow shops, with the objective to either minimize makespan or sum of completion times, have no constant approximation algorithm unless $P = NP$.

Theorem 2. *For all sufficiently large constants K, it is NP-hard to distinguish between generalized flow shop instances that have a schedule with makespan $2K \cdot lb$ and those that have no solution that schedules more than half of the jobs within $(1/8)K^{\frac{1}{25}(\log K)} \cdot lb$ time units. Moreover this hardness result holds for generalized flow shop instances with bounded number of operations per job, that only depends on K.*

By using a similar reduction, but using a stronger assumption, we give a hardness result that essentially shows that the current approximation algorithms for generalized flow shops, with both makespan and sum of weighted completion times objective, are tight.

Theorem 3. *Let $\epsilon > 0$ be an arbitrarily small constant. There is no $O\left((\log lb)^{1-\epsilon}\right)$-approximation algorithm for $F|jumps|C_{max}$ or $F|jumps|\sum C_j$, unless $NP \subseteq ZTIME(2^{O(\log n)^{O(1/\epsilon)}})$.*

No results of this kind were known for a flow shop problem. Moreover, this paper extends and significantly simplifies the recent hardness results by the authors for the acyclic job shop problem [12].

In summary, the consequences of our results are among others that in order to improve the approximation guarantee for flow shops, it is necessary to (i) improve the used lower bound on the optimal makespan and (ii) use the fact that a job needs to be processed on *all* machines.

2 Job and Flow Shops Instances with Large Makespan

We first exhibit an instance of general job scheduling for which it is relatively simple to show that any optimal schedule is of length $\Omega(lb \cdot \log lb)$. The construction builds upon the idea of jobs of different "frequencies", by Feige & Scheideler [4], but we will introduce some important differences that will be decisive for the flow shop case. The resulting instance slightly improves[1] on the bound by Feige & Scheideler [4], who showed the existence of job shop instances with optimal makespan $\Omega(lb \cdot \log lb / \log\log lb)$.

The construction of flow shop instances with "large" makespan is more complicated, as each job is required to have exactly one operation for every machine, and all jobs are required to go through all the machines in the same order. The main idea is to start with the aforementioned job shop construction, which has very cyclic jobs, i.e., jobs have many operations on a single machine. The flow shop instance is then obtained by "copying" the job shop instance several times and, instead of having cyclic jobs, we let the i-th long operation of a job to be processed in the i-th copy of the original job shop instance. Finally, we insert additional zero-length operations to obtain a flow shop instance. We show that the resulting instance has optimal length $\Omega(lb \cdot \log lb / \log\log lb)$.

2.1 Job Shops with Large Makespan

Construction For any integer $d \geq 1$ consider the job shop instance with d machines $m_1, \ldots, m_d$ and d jobs $j_1, \ldots, j_d$. We say that job j_i has *frequency* i, which means that it has 3^i so-called *long-operations* on machine m_i, each one of them requires 3^{d-i} time units. And between any two consecutive long-operations, job j_i has *short-operations* that requires 0 time units on the machines $m_1, \ldots, m_{i-1}$. Note that the length of all jobs and the load on all machines are 3^d, which we denote by lb. For a small example see Figure 1.

Analysis Fix an arbitrarily feasible schedule for the jobs. We shall show that the length of the schedule must be $\Omega(lb \cdot \log lb)$.

Lemma 1. *For $i, j : 1 \leq i < j \leq d$, the number of time units during which both j_i and j_j perform operations is at most $\frac{lb}{3^{j-i}}$.*

[1] However, in their construction all operations of a job have the same length which is not the case for our construction.

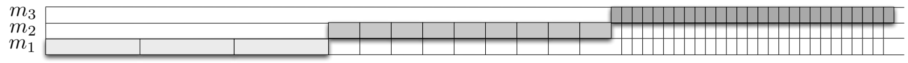

Fig. 1. An example of the construction for job shop with $d = 3$.

Proof. During the execution of a long-operation of j_i (that requires 3^{d-i} time units), job j_j can complete at most one long-operation that requires 3^{d-j} time units (since its short-operation on machine m_i has to wait). As j_i has 3^i long-operations, the two jobs can perform operations at the same time during at most $3^i \cdot 3^{d-j} = \frac{3^d}{3^{j-i}} = \frac{lb}{3^{j-i}}$ time units. $\square$

It follows that, for each $i = 1, \ldots, d$, at most a fraction $1/3+1/3^2+\cdots+1/3^i \leq 1/3+1/3^2+\cdots+1/3^d \leq \frac{1}{3-1} = 1/2$ of the time spent for long-operations of a job j_i is performed at the same time as long-operations of jobs with lower frequency. Hence a feasible schedule has makespan at least $d \cdot lb/2$. As $d = \Omega(\log lb)$ (recall that $lb = 3^d$), the optimal makespan of the constructed job shop instance is $\Omega(lb \cdot \log lb)$.

2.2 Flow Shops with Large Makespan

Construction For sufficiently large integers d and r, consider the flow shop instance defined as follows:

- There are r^{2d} groups of machines[2], denoted by $M_1, M_2, \ldots, M_{r^{2d}}$. Each group M_g consists of d machines $m_{g,1}, m_{g,2}, \ldots, m_{g,d}$ (one for each frequency). Finally the machines are ordered in such a way so that $m_{g,i}$ is before $m_{h,j}$ if either (i) $g < h$ or (ii) $g = h$ and $i > j$. The latter case will ensure that, within each group of machines, long-operations of jobs with high frequency will be scheduled before long-operations of jobs with low frequency, a fact that is used to prove Lemma 3.
- For each frequency $f = 1, \ldots, d$, there are $r^{2(d-f)}$ groups of jobs, denoted by $J_1^f, J_2^f, \ldots, J_{r^{2(d-f)}}^f$. Each group J_g^f consists of r^{2f} copies, referred to as $j_{g,1}^f, j_{g,2}^f, \ldots, j_{g,r^{2f}}^f$, of the job that must be processed during $r^{2(d-f)}$ time units on the machines

  $$m_{a+1,f}, m_{a+2,f}, \ldots, m_{a+r^{2f},f} \text{ where } a = (g-1) \cdot r^{2f}$$

  and during 0 time units on all the other machines that are required to create a flow shop instance. Let J^f be the set of jobs that correspond to frequency f, i.e., $J^f = \{j_{g,a}^f : 1 \leq g \leq r^{2(d-f)}, 1 \leq a \leq r^{2f}\}$.

Note that the length of all jobs and the load on all machines are r^{2d}, which we denote by lb. Moreover, the total number of machines and number of jobs are

[2] These groups of machines "correspond" to copies of the job shop instance in subsection 2.1.

both $r^{2d} \cdot d$. In the subsequent we will call the operations that require more than 0 time units *long-operations* and the operations that only require 0 time units *short-operations*. For an example of the construction see Figure 2.

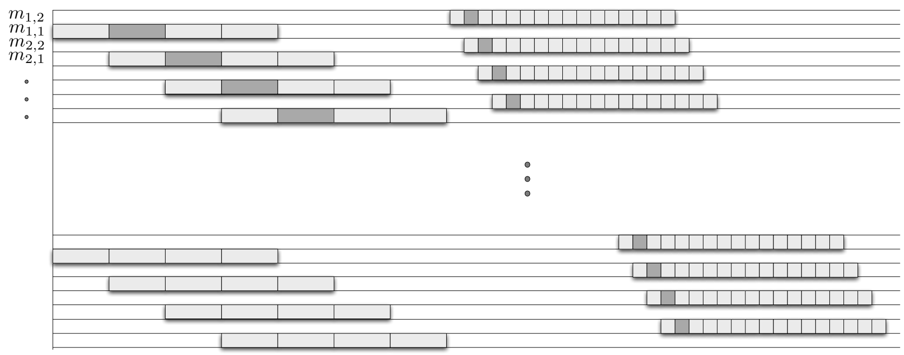

Fig. 2. An example of the construction for flow shop scheduling with $r = d = 2$. Only long-operations on the first 4 and last 4 groups of machines are depicted. The long-operations of one job of each frequency are highlighted in dark gray.

Analysis We shall show that the length of the schedule must be $\Omega(lb \cdot \min[r, d])$. As $lb = r^{2d}$, instances constructed with $r = d$ has optimal makespan $\Omega(lb \cdot \log lb / \log \log lb)$.

Fix an arbitrarily feasible schedule for the jobs. We start by showing a useful property. For a job j, let $d_j(i)$ denote the delay between job j's i-th and $i+1$-th long-operations, i.e., the time units between the end of job j's i-th long-operation and the start of its $i+1$-th long-operation (let $d_j(i) = \infty$ for the last long-operation). We say that the i-th long-operation of a job j of frequency f is *good* if $d_j(i) \leq \frac{r^2}{4} \cdot r^{2(d-f)}$.

Lemma 2. *If the schedule has makespan less than $r \cdot lb$ then the fraction of good long-operations of each job is at least $\left(1 - \frac{4}{r}\right)$.*

Proof. Assume that the considered schedule has makespan less than $r \cdot lb$. Suppose toward contradiction that there exists a job j of frequency f so that j has at least $\frac{4}{r} r^{2f}$ long-operations that are not good. But then the length of j is at least $\frac{4}{r} r^{2f} \cdot \frac{r^2}{4} \cdot r^{2(d-f)} = r \cdot r^{2d} = r \cdot lb$, which contradicts that the makespan of the considered schedule is less than $r \cdot lb$. □

We continue by analyzing the schedule with the assumption that its makespan is less than $r \cdot lb$ (otherwise we are done). In each group M_g of machines we will associate a set $T_{g,f}$ of time intervals with each frequency $f = 1, \ldots, d$. The set $T_{g,f}$ contains the time intervals corresponding to the *first half* of all good long-operations scheduled on the machine $m_{g,f}$.

Lemma 3. *Let $k, \ell : 1 \le k < \ell \le d$ be two different frequencies. Then the sets $T_{g,k}$ and $T_{g,\ell}$, for all $g : 1 \le g \le r^{2d}$, contain disjoint time intervals.*

Proof. Suppose toward contradiction that there exist time intervals $t_k \in T_{g,k}$ and $t_\ell \in T_{g,\ell}$ that overlap, i.e., $t_k \cap t_\ell \neq \emptyset$. Note that t_k and t_ℓ correspond to good long-operations of jobs of frequencies k and ℓ, respectively. Let us say that the good long-operation corresponding to t_ℓ is the a-th operation of some job j. As t_ℓ and t_k overlap, the a-th long-operation of j must overlap the first half of the long-operation corresponding to t_k. As job j has a short operation on machine $m_{g,k}$ after its long-operation on machine $m_{g,\ell}$ (recall that machines are ordered $m_{g,d}, m_{g,d-1}, \ldots, m_{g,1}$ and $\ell > k$), job j's $(a+1)$-th operation must be delayed by at least $r^{2(d-k)}/2 - r^{2(d-\ell)}$ time units and thus $d_j(a) > r^{2(d-k)}/2 - r^{2(d-\ell)} > \frac{r^2}{4} r^{2(d-\ell)}$, which contradicts that the a-th long-operation of job j is good. □

Let $L(T_{g,f})$ denote the total time units covered by the time intervals in $T_{g,f}$. We continue by showing that there exists a g such that $\sum_{f=1}^{d} L(T_{g,f}) \ge \frac{lb}{4} \cdot d$. With this in place, it is easy to see that any schedule has makespan $\Omega(d \cdot lb)$ since all the time intervals $\{T_{g,f} : f = 1, \ldots, d\}$ are disjoint (Lemma 3).

Lemma 4. *There exists a $g \in \{1, \ldots, r^{2d}\}$ such that*

$$\sum_{f=1}^{d} L(T_{g,f}) \ge \frac{lb}{4} \cdot d$$

Proof. As $\sum_{f=1}^{d} L(T_{g,f})$ adds up the time units required by the *first half* of each good long-operation scheduled on a machine in M_g, the claim follows by showing that there exist one group of machines M_g from $\{M_1, M_2, \ldots, M_{r^{2d}}\}$ so that the total time units required by the good long-operations on the machines in M_g is at least $\frac{lb \cdot d}{2}$.

By lemma 2 we have that the good long-operations of each job requires at least $lb \cdot \left(1 - \frac{4}{r}\right)$ time units. Since the total number of jobs is $r^{2d}d$ the total time units required by all good long-operations is at least $lb \cdot \left(1 - \frac{4}{r}\right) \cdot r^{2d} d$. As there are r^{2d} many groups of machines, a simple averaging argument guarantees that in at least one group of machines, say M_g, the total time units required by the good long-operations on the machines in M_g is at least $lb \cdot \left(1 - \frac{4}{r}\right) d > lb \cdot d/2$. □

3 Hardness of Generalized Flow Shops

Theorem 2 and Theorem 3 are proved by presenting a gap-preserving reduction Γ from the graph coloring problem to the generalized flow shop problem. Γ has two parameters r and d. Given an n-vertex graph G whose vertices are partitioned into d independent sets, it computes in time polynomial in n and r^d, a generalized flow shop instance $S(r, d)$ where all jobs have the same length r^{2d} and all machines the same load r^{2d}. Hence, $lb = r^{2d}$. Instance $S(r, d)$ has a set of r^{2d} jobs and and a set of r^{2d} machines for each vertex in G. The total number of jobs and the total number of machines are thus both $r^{2d}n$. Moreover, each job has at most $(\Delta + 1)r^{2d}$ operations. By using jobs of different frequencies, as done in the gap construction, we have the property that "many" of the jobs corresponding to adjacent vertices cannot be scheduled in parallel in any feasible schedule. On the other hand, by letting jobs skip those machines corresponding to non-adjacent vertices, jobs corresponding to an independent set in G *can* be scheduled in parallel (their operations can overlap in time) in a feasible schedule. This ensures that the following completeness and soundness hold for the resulting generalized flow shop instance $S(r, d)$.

- Completeness case: If G can be colored using L colors then $C^*_{max} \leq lb \cdot 2L$;
- Soundness case: For any $L \leq r$. Given a schedule where at least half the jobs finish within $lb \cdot L$ time units, we can, in time polynomial in n and r^d, find an independent set of G of size $n/(8L)$.

In Section 3.1 we describe the gap-preserving reduction Γ. With this construction in place, Theorem 2 easily follows by using a result by Khot [8], that states that it is NP-hard to color a K-colorable graph with $K^{\frac{1}{25}(\log k)}$ colors, for sufficiently large constants K. The result was obtained by presenting a polynomial time reduction that takes as input a SAT formula ϕ together with a sufficiently large constant K, and outputs an n-vertex graph G with degree at most $2^{K^{O(\log K)}}$. Moreover, (completeness) if ϕ is satisfiable then graph G can be colored using K colors and (soundness) if ϕ is not satisfiable then graph G has no independent set containing $n/K^{\frac{1}{25}(\log K)}$ vertices (see Section 6 in [8]). Note that the soundness case implies that any feasible coloring of the graph uses at least $K^{\frac{1}{25}(\log K)}$ colors. We let $\mathcal{G}[c, i]$ be the family of graphs that either can be colored using c colors or have no independent set containing a fraction i of the vertices. To summarize, for sufficiently large K and $\Delta = 2^{K^{O(\log K)}}$, it is NP-hard to decide if an n-vertex graph G in $\mathcal{G}[K, 1/K^{\frac{1}{25}(\log K)}]$ with bounded degree Δ has

$$\chi(G) \leq K \text{ or } \alpha(G) \leq \frac{n}{K^{\frac{1}{25}(\log K)}}$$

where $\chi(G)$ and $\alpha(G)$ denote the chromatic number and the size of a maximum independent set of G, respectively. As the vertices of a graph with bounded degree Δ can, in polynomial time, be partitioned into $\Delta + 1$ independent sets, we can use Γ with parameters $d = \Delta+1$ and $r = K^{\frac{1}{25}(\log K)}$ (r is chosen such that the condition $L \leq r$ in the soundness case of Γ is satisfied for $L = K^{\frac{1}{25}(\log K)}/8$). It follows, by the completeness case and soundness case of Γ, that it is NP-hard to distinguish if the obtained scheduling instance has a schedule with makespan at most $lb \cdot 2K$, or no solution schedules more than half of the jobs within $lb \cdot K^{\frac{1}{25}(\log K)}/8$ time units. Moreover, each job has at most $(\Delta+1)r^{2d}$ operations, which is a constant that only depends on K.

The proof of Theorem 3 is similar to the proof of Theorem 2 with the exception that the graphs have no longer bounded degree. Due to space limits the proof is omitted; it follows the one provided by the authors for the acyclic job shop problem [12].

3.1 Construction

Here, we present the reduction Γ for the general flow shop problem where jobs are allowed to skip machines. Given an n-vertex graph $G = (V, E)$ whose vertices are partitioned into d independent sets, we create a generalized flow shop instance $S(r, d)$, where r and d are the parameters of the reduction. Let $I_1, I_2, \ldots I_d$ denote the independent sets that form a partition of V.

$S(r, d)$ is very similar to the gap instance described in Section 2.2. The main difference is that in $S(r, d)$ distinct jobs can be scheduled in parallel if their corresponding vertices in G are not adjacent. This is obtained by letting a job to skip those machines corresponding to non-adjacent vertices. (The gap instance of Section 2.2 can be seen as the result of the following reduction when the graph G is a complete graph with d nodes). For convenience, we give the complete description with the necessary changes.

- There are r^{2d} groups of machines, denoted by $M_1, M_2, \ldots, M_{r^{2d}}$. Each group M_g consists of n machines $\{m_{g,v} : v \in V\}$ (one for each vertex in G). Finally the machines are ordered in such a way so that $m_{g,u}$ is before $m_{h,v}$ if either (i) $g < h$ or (ii) $g = h$ and $u \in I_k, v \in I_\ell$ with $k > \ell$. The latter case will ensure that, within each group of machines, long-operations of jobs with high frequency will be scheduled before long-operations of jobs with low frequency, a fact that is used to prove Lemma 8.
- For each $f : 1 \leq f \leq d$ and for each vertex $v \in I_f$ there are $r^{2(d-f)}$ groups of jobs, denoted by $J_1^v, J_2^v, \ldots, J_{r^{2(d-f)}}^v$. Each group J_g^v consists of r^{2f} copies, referred to as $j_{g,1}^v, j_{g,2}^v, \ldots, j_{g,r^{2f}}^v$, of the job that must be processed during $r^{2(d-f)}$ time units on the machines

  $$m_{a+1,v}, m_{a+2,v}, \ldots, m_{a+r^{2f},v} \text{ where } a = (g-1) \cdot r^{2f}$$

  and during 0 time units on machines corresponding to adjacent vertices, i.e., $\{m_{a,u} : 1 \leq a \leq r^{2d}, \{u, v\} \in E\}$ in an order such that it results in a generalized flow shop instance. Let J^v be the jobs that correspond to the vertex v, i.e., $J^v = \{j_{g,i}^v : 1 \leq g \leq r^{2(d-f)}, 1 \leq i \leq r^{2f}\}$.

Note that the length of all jobs and the load on all machines are r^{2d}, which we denote by *lb*. The total number of machines and total number of jobs are both $r^{2d} \cdot n$. Moreover, each job has at most $(\Delta+1)r^{2d}$ operations. In the subsequent we will call the operations that require more than 0 time units *long-operations* and the operations that only require 0 time units *short-operations*. For an example of the construction see Figure 3.

Completeness We prove that if the graph G can be colored with "few" colors then there is a relatively "short" schedule to the general flow shop instance.

Lemma 5. *There is a schedule of $S(r, d)$ with makespan $2lb \cdot \chi(G)$.*

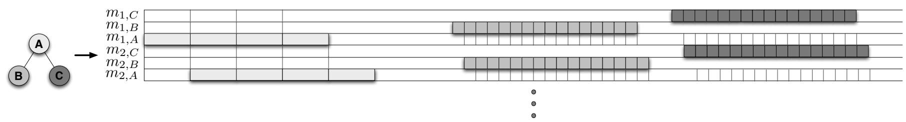

Fig. 3. An example of the reduction with $r = 2, d = 2, I_1 = \{A\}$ and $I_2 = \{B, C\}$. Only the two first out of $r^{2d} = 16$ groups of machines are depicted with the jobs corresponding to A,B, and C to the left, center, and right respectively.

Proof. We start by showing that all jobs corresponding to non-adjacent vertices can be scheduled within $2 \cdot lb$ time units.

Claim. Let IS be an independent set of G. Then all the jobs $\bigcup_{v \in IS} J^v$ can be scheduled within $2 \cdot lb$ time units.

Proof of Claim. Consider the schedule defined by scheduling the jobs corresponding to each vertex $v \in IS$ as follows. Let I_f be the independent set with $v \in I_f$. A job $j^v_{g,i}$ corresponding to vertex v is then scheduled without interruption starting from time $r^{2(d-f)} \cdot (i - 1)$.

The schedule has makespan at most $2 \cdot lb$ since a job is started at latest at time $r^{2(d-f)} \cdot (r^{2f} - 1) < lb$ and requires lb time units in total.

To see that the schedule is feasible, observe that no short-operations of the jobs in $\bigcup_{v \in IS} J^v$ need to be processed on the same machines as the long-operations of the jobs in $\bigcup_{v \in IS} J^v$ (this follows from the construction and that the jobs correspond to non-adjacent vertices). Moreover, two jobs $j^v_{g,i}, j^{v'}_{g',i'}$, with either $g \neq g'$ or $v \neq v'$, have no two long-operations that must be processed on the same machine. Hence, the only jobs that might delay each other are jobs belonging to the same vertex v and the same group g, but these jobs are started with appropriate delays (depending on the frequency of the job). □

We partition set V into $\chi(G)$ independent subsets $V_1, V_2, \dots, V_{\chi(G)}$. By the above lemma, the jobs corresponding to each of these independent sets can be scheduled within $2 \cdot lb$ time units. We can thus schedule the jobs in $\chi(G)$-"blocks", one block of length $2 \cdot lb$ for each independent set. The total length of this schedule is $2lb \cdot \chi(G)$. □

Soundness We prove that, given a schedule where many jobs are completed "early", we can, in polynomial time, find a "big" independent set of G.

Lemma 6. *For any $L \leq r$. Given a schedule of $S(r, d)$ where at least half the jobs finish within $lb \cdot L$ time units , we can, in time polynomial in n and r^d, find an independent set of G of size at least $n/(8L)$.*

Proof. Fix an arbitrarily schedule of $S(r, d)$ where at least half the jobs finish within $lb \cdot L$ time units. In the subsequent we will disregard the jobs that do not finish within $lb \cdot L$ time units throughout the analysis. Note that the remaining jobs are at least $r^{2d}n/2$ many. As for the gap construction (see Section 2.2), we say that the i-th long-operation of a job j of frequency f is *good* if the delay $d_j(i)$ between job j's i-th and $i+1$-th long-operations is at most $\frac{r^2}{4} \cdot r^{2(d-f)}$. In each group M_g of machines we will associate a set $T_{g,v}$ of time intervals with each vertex $v \in V$. The set $T_{g,v}$ contains the time intervals corresponding to the *first half* of all good long-operations scheduled on the machine $m_{g,v}$. We also let $L(T_{g,v})$ denote the total time units covered by the time intervals in $T_{g,v}$. Scheduling instance $S(r, d)$ has a similar structure as the gap instances created in Section 2.2 and has similar properties. By using the fact that all jobs (that were not disregarded) have completion time at most $L \cdot lb$ which is by assumption at most $r \cdot lb$, Lemma 7 follows from the same arguments as Lemma 2.

Lemma 7. *The fraction of good long-operations of each job is at least* $\left(1 - \frac{4}{r}\right)$.

Consider a group M_g of machines and two jobs corresponding to adjacent vertices that have long-operations on machines in M_g. Recall that jobs corresponding to adjacent vertices have different frequencies. By the ordering of the machines, we are guaranteed that the job of higher frequency has, after its long-operation on a machine in M_g, a short-operation on the machine in M_g where the job of lower frequency has its long-operation. The following lemma now follows by observing, as in the proof of Lemma 3, that the long-operation of the high frequency job can only be good if it is *not* scheduled in parallel with the first half of the long-operation of the low frequency job.

Lemma 8. *Let $u \in I_k$ and $v \in I_l$ be two adjacent vertices in G with $k > l$. Then the sets $T_{g,u}$ and $T_{g,v}$, for all $g : 1 \leq g \leq r^{2d}$, contain disjoint time intervals.*

Finally, Lemma 9 is proved in the very same way as Lemma 4. Their different inequalities arise because in the gap instance we had $d \cdot r^{2d}$ jobs and here we are considering at least $r^{2d}n/2$ jobs that were not disregarded.

Lemma 9. *There exists a $g \in \{1, \ldots, r^{2d}\}$ such that*

$$\sum_{v \in V} L(T_{g,v}) \geq \frac{lb \cdot n}{8}.$$

We conclude by a simple averaging argument. Set g so that $\sum_{v \in V} L(T_{g,v})$ is at least $\frac{lb \cdot n}{8}$, such a g is guaranteed to exist by the lemma above. As all jobs that were not disregarded finish within $L \cdot lb$ time units, at least $\frac{lb \cdot n}{8}/(L \cdot lb) = \frac{n}{8L}$ time intervals must overlap at some point during the first $L \cdot lb$ time units of the schedule, and, since they overlap, they correspond to different vertices that form an independent set in G (Lemma 8). Moreover, we can find such a point in the schedule by, for example, considering all different blocks and, in each block, verify the start and end points of the time intervals. □

Acknowledgments.

This research is supported by the Swiss National Science Foundation project "Approximation Algorithms for Machine scheduling Through Theory and Experiments III" Project N. 200020-122110/1.

References

1. J. Beck. An algorithmic approach to the lovasz local lemma. *Random Structures and Algorithms*, 2(4):343–365, 1991.
2. B. Chen, C. Potts, and G. Woeginger. A review of machine scheduling: Complexity, algorithms and approximability. *Handbook of Combinatorial Optimization*, 3:21–169, 1998.
3. A. Czumaj and C. Scheideler. A new algorithm approach to the general lovász local lemma with applications to scheduling and satisfiability problems (extended abstract). In *STOC*, pages 38–47, 2000.
4. U. Feige and C. Scheideler. Improved bounds for acyclic job shop scheduling. *Combinatorica*, 22(3):361–399, 2002.
5. L. Goldberg, M. Paterson, A. Srinivasan, and E. Sweedyk. Better approximation guarantees for job-shop scheduling. *SIAM Journal on Discrete Mathematics*, 14(1):67–92, 2001.
6. R. Graham, E. Lawler, J. Lenstra, and A. R. Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. In *Annals of Discrete Mathematics*, volume 5, pages 287–326. North–Holland, 1979.
7. H. Hoogeveen, P. Schuurman, and G. J. Woeginger. Non-approximability results for scheduling problems with minsum criteria. *INFORMS Journal on Computing*, 13(2):157–168, 2001.
8. S. Khot. Improved inapproximability results for maxclique, chromatic number and approximate graph coloring. In *FOCS*, pages 600–609, 2001.
9. E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys. Sequencing and scheduling: Algorithms and complexity. *Handbook in Operations Research and Management Science*, 4:445–522, 1993.
10. F. T. Leighton, B. M. Maggs, and S. B. Rao. Packet routing and job-shop scheduling in O(congestion + dilation) steps. *Combinatorica*, 14(2):167–186, 1994.
11. F. T. Leighton, B. M. Maggs, and A. W. Richa. Fast algorithms for finding O(congestion + dilation) packet routing schedules. *Combinatorica*, 19:375–401, 1999.
12. M. Mastrolilli and O. Svensson. (Acyclic) jobshops are hard to approximate. In *FOCS*, pages 583–592, 2008.
13. V. Nagarajan and M. Sviridenko. Tight bounds for permutation flow shop scheduling. In *IPCO*, pages 154–168, 2008.
14. C. Potts, D. Shmoys, and D. Williamson. Permutation vs. nonpermutation flow shop schedules. *Operations Research Letters*, 10:281–284, 1991.
15. M. Queyranne and M. Sviridenko. Approximation algorithms for shop scheduling problems with minsum objective. *Journal of Scheduling*, 5(4):287–305, 2002.
16. P. Schuurman and G. J. Woeginger. Polynomial time approximation algorithms for machine scheduling: ten open problems. *Journal of Scheduling*, 2(5):203–213, 1999.
17. D. Shmoys, C. Stein, and J. Wein. Improved approximation algorithms for shop scheduling problems. *SIAM Journal on Computing*, 23:617–632, 1994.
18. D. Williamson, L. Hall, J. Hoogeveen, C. Hurkens, J. Lenstra, S. Sevastianov, and D. Shmoys. Short shop schedules. *Operations Research*, 45:288–294, 1997.