Makespan minimization in job shops: a polynomial time approximation scheme

M.I. Sviridenko
Sobolev Institute of Mathematics
e-mail: svir@math.nsc.ru

Abstract

In this work we obtain a polynomial time approximation scheme for the job shop problem with fixed numder of machines, with fixed maximum number of operations per job and the minimum makespan objective.

1 Introduction

In the job shop problem n jobs have to be processed on m different machines. Each job consists of a sequence of operations that have to be processed in a fixed order specified by this sequence. An operation has to be executed on a given machine during a fixed processing time without interruptions. The aim is to minimize the maximum of the job completion times (makespan).

We formally define the job shop scheduling problem as follows. We are given a set $M = \{M_1, \ldots, M_m\}$ of machines and a set $J = \{J_1, \ldots, J_n\}$ of jobs. Job J_j consists of μ_j operations $\{O_{1j}, \ldots, O_{\mu_j j}\}$ that have to be processed in this order. Operation O_{ij} must be processed without preemtions on machine M_t for some $t = t_{ij}$, which requires p_{ij} time units. Each machine can process at most one operation at time, and each job may be processed by at most one machine at a time. Let C_{ij} stand for the completion time of operation O_{ij}. The objective is to produce a schedule that minimizes the maximum completion time, $C_{max} = \max_{i,j} C_{ij}$; the optimal value of C_{max} is denoted by C^*_{max}. Define $P_t = \sum_{t_{ij}=t} p_{ij}$ and $P = \max\{P_1, \ldots, P_m\}$. In other words, P is the maximum machine load. Clearly, $P \leq C^*_{max}$.

It is well-known that the job shop problem is a one of the most intractable problems in combinatorial optimization both from theoretical and practical viewpoints. Almost all special cases are strongly NP-hard (see survey [6]). Williamson et al.[14] proved that for any $\rho < 5/4$, the existence of a ρ-approximation algorithm for any shop scheduling problem with the number of machines as part of the input would imply $P = NP$.

During the last few years, many papers devoted to approximating of various job shop problems appeared. Almost all these papers are based on the seminal work by Leighton, Maggs and Rao [7]. The main result of Leighton, Maggs and Rao was to show that there always exists a schedule of length $O(P + l_{max})$ for the acyclic job shop problem with operations of unit length where $l_{max} = \max_j \sum_i p_{ij}$ is the maximum job length. This is not an algorithmic result since it relies on a nonconstructive probabilistic argument (for construcive version see [8]). Shmoys, Stein and Wein [13] described an approximation algorithm for the general job shop problem. The performance guarantee of this algorithm was $O(\log^2(m\mu))$. This algorithm was slightly improved by Goldberg, Paterson, Srinivasan and Sweedyk [1]. Shmoys, Stein and Wein [13] also proposed a polynomial time approximation algorithm with performance ratio of $(2+\varepsilon)$ for any fixed $\varepsilon > 0$ for the job shop problem with fixed m and μ. Hall [2] developed a polynomial time approximation scheme (PTAS) for the flow shop problem with fixed number of machines.

In this work we describe a PTAS for the job shop problem with fixed m and μ. In fact, our scheme can be applied for substantially more general problems (see section 3). For example, our scheme works for a co-called dag shop problem, where the order of processing the operations of a job must be consistent with a given partial order. In particular, this problem includes the open shop problem. And therefore our result generalizes recent result on existence PTAS for the open shop problem due to Sevastianov and Woeginger [12] (of course, the running time of their PTAS is much better).

Our work is strongly based on ideas of the papers mentioned above. We use the idea by Sevastianov and Woeginger [12] of dividing the set of operations into three subsets of big, small and tiny operations, we use the idea by Hall [2] of using a linear programming, the idea of apllication Sevastianov's [10] algorithm for scheduling tiny jobs due to Shmoys, Stein and Wein [13] and, of course, the seminal algorithm by Sevastianov [10]. More precisely, Sevastianov obtained an algorithm for the job shop problem with running time $O((\mu mn)^2)$ which always finds a schedule of length at most $P + \varphi(m, \mu)p_{max}$, where $\varphi(m, \mu) = (m\mu^2 + 2\mu - 1)(\mu - 1) = O(m\mu^3)$ and $p_{max} = \max_{i,j} p_{ij}$ is the maximum operation length. (See also [11] for survey and historical overview of geometric methods in design and analysis of approximation algorithms with absolute performance guarantee for scheduling problems.)

Our PTAS is also based on the notion of an *outline scheme* introduced by Hall and Shmoys [3, 4, 5] and Hall [2]. Hall [2] gave the following definition of an outline scheme:

> An outline scheme is a partition of all feasible schedules into sets, whereby the schedules are grouped together into a set according to some characteristics that they share. The goal is that two things are simultaneously possible: first, that the partition has at most a polynomial number of sets in it; and second, that, given characteristics describing a set of the partition, it is possible to generate a schedule in polynomial time that is nearly as good as any schedule in that set. The outline scheme then suggests a natural way to obtain a good schedule: simply generate, for each possible set in the partition, a schedule that

is "good" relative to that set. Since the optimal schedule must be contained in one of the sets, we are guaranteed to find a schedule that is " nearly" optimal, among all of the schedules generated.

2 Algorithm and its analysis

Let ε be an arbitrary small rational number in the interval $(0,1)$. Let $m \geq 2$ and $\mu \geq 1$ be integers. We will assume that the values of ε, μ and m are fixed and not part of the input. Following [12], we partition the set of operations into three subsets. For a real number α with

$$(\varepsilon)^{2^{\lceil m/\varepsilon \rceil}} \leq \alpha \leq \varepsilon,$$

we define three sets of operations:

$$B = \{O_{ij} | p_{ij} \geq \alpha P\},$$

$$S = \{O_{ij} | \alpha^2 P < p_{ij} < \alpha P\},$$

$$T = \{O_{ij} | p_{ij} \leq \alpha^2 P\}.$$

The operations in B are called *big operations* (or B-operations), the operations in S are called *small operations* (or S-operations) and operations in T are called the *tiny operations* (or T-operations). Partition the tiny operations into two sets: $T = T_1 \cup T_2$, where T_1 is the set of T-operations of jobs that have at least one small or big operation and T_2 is the set of T-operations of jobs consisting of T-operations only. The operations in T_i are called T_i-operations for $i = 1, 2$. The number of B-operations is bounded by a constant depending only on ε and m:

$$|B| \leq m/\alpha \leq m(\varepsilon)^{-2^{\lceil m/\varepsilon \rceil}}.$$

Sevastianov and Woeginger [12] showed that we can choose the number α so that

$$\sum_{O_{ij} \in S} p_{ij} \leq \varepsilon P. \qquad (1)$$

This was done as follows. Define a sequence of numbers α_l, where l is a nonnegative integer, by $\alpha_l = (\varepsilon)^{2^l}$ and consider the sets S_l of small operations with respect to α_l. The sets S_i and S_j are disjoint for $i \neq j$. Since the total length of all operations is at most mP, there exists $k \leq \lceil m/\varepsilon \rceil$ for which S_k fulfills inequality (1). We set $\alpha = \alpha_k$.

The number mP is an upper bound on the length of the optimal schedule. We partition the time interval from 0 to $m(1+\alpha)P$ into $\lceil m(1+\alpha)/\alpha^2 \rceil$ intervals of length $\alpha^2 P$. These intervals are called *the intervals of the first type* .

We will consider schedules such that every B-operation start its processing at the beginning of some interval of the first type. It is easy to see that such a restriction does not increase the length of the optimal schedule considerably. Indeed, let us consider the first B-operation at the optimal schedule starting not in the beginning of some interval. We can simply shift this B-operation to the right, so that it starts at the beginning of the next interval. All operations starting after this B-operation are shifted to the right by the same length. The overall increase of the length of the optimal schedule is bounded by $|B|\alpha^2 P \leq m\alpha P$. Let $\tilde{C}^*_{max}$ be the length of the optimal schedule of the restricted problem. As was noted above $\tilde{C}^*_{max} \leq C^*_{max} + m\alpha P$. Further we will assume that every B-operation starts at the beginning of some interval of the first type.

Let us fix some feasible schedule for B-operations within the time interval $[0, m(1+\alpha)P]$. Let S_{ij} and C_{ij} be the starting and the completion time of operation O_{ij}, respectively. Let $A = \{a | a = S_{ij} \text{ or } a = C_{ij} \text{ for some operation } O_{ij} \in B\}$. In other words, A is the set of starting and completion times of B-operations. Notice that $|A| = 2|B| \leq 2m/\alpha$. Assume that the set A is numerated so that $a_1 \leq a_2 \leq \ldots \leq a_{|A|}$. We add two elements $a_0 = 0$ and $a_{|A|+1} = C \in [a_{|A|}, (1+\alpha)mP]$ (the exact value of C will be specified later) to the set A and partition the time interval from 0 to C into $|A|+1$ intervals $[a_k, a_{k+1}), k = 0, \ldots, |A|$. We will call this intervals *the intervals of the second type.* Let Δ_k be the length of the kth interval, i.e. $\Delta_k = a_{k+1} - a_k$.

For each S-operation and each T_1-operation we fix the time interval of the second type in which that operation starts its processing on its associated machine.

We have described an outline scheme. Since the number of big, small and T_1-operations is constant and since the number of intervals of both types is constant, we deduce that the number of possible groups of schedules (outlines) with fixed starting times of B-operations and fixed starting intervals of S-operations and T_1-operations is constant, too. For each such outline we assign the starting times of T_2-operations to some interval of the second type by means of linear programming.

Let S^{tk} and T_1^{tk} be the sets of small and T_1-operations with starting times scheduled to the kth interval on machine M_t, let I be the set of indices of jobs consisting on the T_2-operations only, and let

$$K_j = \{K = (k_1, k_2, \ldots, k_{\mu_j}) \in Z^{\mu_j} |\ 0 \leq k_1 \leq k_2 \leq \ldots \leq k_{\mu_j} \leq |A|\}$$

be the set of all feasible assignments of starting times of operations of job $J_j, j \in I$ to the intervals of the second type. Now we construct a linear program with decision variables

$$x_{j,K}, \quad j \in I, \quad K \in K_j,$$

where $x_{j,K} = 1$ means that job J_j gets starting time of its first operation assigned to interval k_1 on machine $M_t, t = t_{1j}$, starting time of the second operation assigned to interval k_2 on $M_t, t = t_{2j}$ and so on. Let

$$\Delta'_{tk} = \Delta_k - \sum_{O_{ij} \in S^{tk} \cup T_1^{tk}} p_{ij} + d(k)$$

where

$$d(k) = \max\{\alpha^2 P, \max_{O_{ij} \in \cup_{t=1}^m S^{tk}} p_{ij}\}.$$

In other words, Δ'_{tk} is an upper bound on free time in interval k on machine M_t remained after assigning S-operations and T_1-operations (we added the last summand because one small or tiny operation may start in the schedule at the very end of the interval). We wish to find a basis feasible solution to the following linear program:

$$\sum_{K \in K_j} x_{j,K} = 1, \quad j \in I, \tag{2}$$

$$\sum_{j \in I} \sum_{K \in K_j} P_{j,K,t,k} \cdot x_{j,K} \leq \Delta'_{tk}, \quad t = 1, \ldots, m, \; k = 0, \ldots, |A|, \tag{3}$$

$$x \geq 0, \tag{4}$$

where $P_{j,K,t,k}$ is the total processing time over all operations O_{ij} of job J_j such that $t_{ij} = t$ and $k_i = k$. Less formally, constraint (3) means that the total length of jobs assigned to interval k on machine M_t does not exceed the length of the interval.

Recall that $a_{|A|+1} = C \in [a_{|A|}, (1+\alpha)mP]$. Let C_{min} be the smallest C such that linear program (2)-(4) has a feasible solution for some outline. Since for $C = \tilde{C}^*_{max}$ we can construct such outline by means of optimal solution of constrained job shop problem (problem with fixed starting times of B-operations) we deduce $C_{min} \leq \tilde{C}^*_{max}$. For any $\delta \geq 0$, the value of C satisfying inequalities $C_{min} \leq C \leq C_{min} + \delta$ can be found in polynomial time by binary search. Since C_{min} is a lower bound on $\tilde{C}^*_{max}$ we obtain $C \leq C^*_{max} + m\alpha P + \delta \leq C^*_{max} + 2m\alpha P$ for $\delta = m\alpha P$.

Linear program (2)-(4) has $|I| + m(|A|+1)$ constraints and $|I|(|A|+1)^\mu$ variables. Therefore, a basis solution of this program is guaranteed to have at most $|I| + m(|A|+1)$ nonzero variables. This solution can have at most $m(|A|+1)$ fractional variables, since each job has at least one positive variable associated with it. (This argument was first made and exploited by Potts [9] in context of parallel machine scheduling.) We will schedule these jobs at the end of the algorithm. The remaining T_2-operations have a unique assignment to the intervals of the second type.

Assume that all small and tiny operations are scheduled to begin in some interval of the second type. Consider interval $[a_k, a_{k+1})$. Let $p_{max}(k)$ be the length of the longest small or tiny operation assigned to this interval. By construction, the total length of operations assigned to this interval on each machine is at most $a_{k+1} - a_k + d(k)$. We consider the problem of scheduling the small and tiny operations in interval k independently of other intervals. Hence, we have a small job shop problem, and by means of Sevastianov's algorithm we can find a feasible schedule of length at most $a_{k+1} - a_k + d(k) + O(m\mu^3)p_{max}(k)$. If we add $d(k) + O(m\mu^3)p_{max}(k)$ to the length of each interval, we can obtain a feasible schedule for the original instance of the constrained job shop problem where we assume that all B-operations start at the beginning of the prescribed interval. The makespan of this schedule is at most

$$C + \sum_{k=0}^{|A|} d(k) + O(m\mu^3) \sum_{k=0}^{|A|} p_{max}(k) \leq C + O(m\mu^3)(\sum_{O_{ij} \in S} p_{ij} + \sum_{k=0}^{|A|} \alpha^2 P) \leq$$

$$C + O(m\mu^3)\varepsilon P + O(m^2\mu^3)\alpha P \leq C^*_{max} + O(m^2\mu^3)\varepsilon P.$$

After that we independently schedule $m(|A|+1)$ jobs corresponding to fractional variables in the solution of the program (2)-(4) by a greedy algorithm (scheduling each operation as early as possible), we increase makespan by at most

$$\mu m(|A|+1) \cdot \alpha^2 P \leq O(m^2\mu)\alpha P.$$

Since m and μ are both constants and ε is an arbitrary rational number we have a PTAS built for the $Jm|\cdot|C_{max}$ with fixed μ.

3 Generalizations

First we consider a simple generalization to multiprocessor job shop problem in which the number of parallel machines at any stage is also fixed. The multiprocessor job shop problem is one in which each machine is replaced by a set of m_i parallel identical machines, $i = 1, \ldots, m$. If all m_i are fixed, our scheme needs no change: we schedule B-operations to start at the beginning of intervals of the first type (the intervals are defined on each machine). After that, we schedule small and T_1-operations to begin in some interval of the second type, and we schedule T_2-operations by using linear programming. Since Sevastianov's algorithm works in the case of multiprocessor job shops (even in the case of multiprocessor dag shop problem), we obtained an approximation scheme for it.

It is interesting that scheme by Hall can be simply generalized to the case of flow shops with a nonbotleneck machine which is equivalent to an infinite collection of identical parallel machines. But we cannot directly extend our algorithm to this case, because Sevastianov's algorithm does not work with nonbotleneck machines. However, our scheme can be generalized to the job shop problem with release dates. It can be done as follows. Partition the time interval from 0 to $R + m(1+\alpha)P$ (R is a maximal release date) into intervals of the length $\alpha \max\{R, P\}$ and round each release date up to the nearest end of the interval. We will obtain at most $(m(1+\alpha)+1)/\alpha$ different release dates and lengthen the optimal schedule by at most $\alpha \max\{R, P\}$. We set $A = \{a | a = S_{ij}$ or $a = C_{ij}$ for $O_{ij} \in B$ or $a = r_k, k = 1, \ldots, (m(1+\alpha)+1)/\alpha\}$ where r_k is rounded release dates. Then the number of intervals of the second type will be $O(m/\alpha)$. And we can apply the rest part of our algorithm without changes.

Acknowledgments

We are grateful to Alexander Kononov and Sergey Sevastianov for many helpful discussions and comments.

References

[1] L. A. Goldberg, M. Paterson, A Srinivasan and E. Sweedyk, Better approximation guarantees for job-shop scheduling, In Proceedings of the 8th Symposium on Discrete Algorithms (1997), pp.599–608.

[2] L. A. Hall, Approximability of flow shop scheduling, Mathematical Programming **82** (1998), pp.175–190.

[3] L. A. Hall and D. B. Shmoys, Approximation algorithms for constrained scheduling problems, In Proceedings of the IEEE 30th Annual Symposium on Foundations of Computer Science (1989), pp.134–139.

[4] L. A. Hall and D. B. Shmoys, Near-optimal sequencing with precedence constraints, In Proceedings of IPCO (1990), pp.249–260.

[5] L. A. Hall and D. B. Shmoys, Jackson's rule for single -machine scheduling: Making a good heuristic better, Mathematics of Operation Research **17** (1992), pp.22–35.

[6] E. L. Lawler, J. K. Lenstra, A. H. G. Rinooy Kan and D. B. Shmoys, Sequencing and scheduling: Algorithms and complexity, In S. C. Graves, A. H. G. Rinooy Kan and P. H. Zipkin, editors, Handbook in Operation Research and Management Science, **v. 4**, Logistics of Production and Inventory, North-Holland (1993), pp.445–522.

[7] T. Leighton, B. Maggs and S. Rao, Packet routing and job-shop scheduling in O(Congestion+Dilation) Steps, Combinatorica **14** (1994), pp.167–186.

[8] T. Leighton, B. Maggs and A. Richa, Fast algorithms for finding O(congestion+dilation) packet routing schedules, Technical Report CMU-CS-96-152, School of Computer Science, Carnegie Mellon University (1996).

[9] C. N. Potts, Analysis of a linear programming heuristic for scheduling unrelated parallel machines, Discrete Applied Mathematics **10** (1985), pp.155–164.

[10] S. V. Sevast'janov, Bounding algorithm for the routing problem with arbitrary paths and alternative servers, Cybernetics **22** (1986), pp.773–780.

[11] S. V. Sevast'janov, On some geometric methods in scheduling theory: a survey, Discrete Applied Mathematics **55** (1994), pp.59–82.

[12] S. V. Sevastianov, G. J. Woeginger, Makespan minimization in open shops: A polynomial time approximation scheme, Mathematical Programming **82** (1998), pp.191–198.

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

[14] D. P. Williamson, L. A. Hall, J. A. Hoogeveen, C. A. J. Hurkens, J. K. Lenstra, S. V. Sevast'janov and D. B. Shmoys, Short shop schedules, Operation Research **45** (1997), pp.288–294.