A Randomized Algorithm for Flow Shop Scheduling

Naveen Garg[1], Sachin Jain[2], and Chaitanya Swamy[3]

[1] Department of Computer Science, Indian Institute of Technology,
Hauz Khas, New Delhi - 110016, India
naveen@cse.iitd.ernet.in

[2] Department of Computer Sciences, The University of Texas at Austin,
Austin, TX 78712-1188, USA
sachin@cs.utexas.edu

[3] Department of Computer Science, Cornell University, Ithaca, NY 14853, USA
swamy@cs.cornell.edu

Abstract. Shop scheduling problems are known to be notoriously intractable, both in theory and practice. In this paper we give a randomized approximation algorithm for flow shop scheduling where the number of machines is part of the input problem. Our algorithm has a multiplicative factor of $2(1+\delta)$ and an additive term of $O(m \ln(m+n) p_{max})/\delta^2)$.

1 Introduction

Shop scheduling has been studied extensively in many varieties. The basic shop scheduling model consists of *machines* and *jobs* each of which consists of a set of *operations*. Each operation has an associated machine on which it has to be processed for a given length of time. The processing times of operations of a job cannot overlap. Each machine can process at most one operation at a given time. We assume that there are m machines and n jobs. The processing time(of an operation) of job j on machine i is denoted by p_{ij} and $p_{\max} = \max p_{ij}$. We will use the terms job(operation) size and processing time interchangeably.

The three well-studied models are the open shop, flow shop and job shop problems. In an open shop problem, the operations of a job can be performed in any order; in a job shop, they must be processed in a specific, job-dependent order. A flow shop is a special case of job shop in which each job has exactly m operations – one per machine, and the order in which they must be processed is same for all the jobs. The problem is to minimize the makespan, ie. the overall length, of the schedule with the above constraints.

All the above problems are strongly **NP**-Hard in their most general form. For job shops, extremely restricted versions are also strongly **NP**-Hard; for example when there are two machines and the all operations have processing times of one or two time units. For the flow shop problem, the case when there are more than two machines is strongly **NP**-Hard, although the two machine version is polynomially solvable [1]. The open shop problem is weakly **NP**-Hard when the number of machines is fixed(but arbitrary) and its relation to being strongly **NP**-Hard is open. For two machines there exists a polynomial algorithm for open shops.

As far as approximability of these models is concerned, Williamson et al. [2] proved a lower bound 5/4 for the problems in their most general form. For the general open shop problem a greedy heuristic is a 2-approximate algorithm. In the case of job shops and flow shops, an algorithm by Sevast'janov [3] gives an additive approximation of $m(m-1)p_{max}$. Shmoys, Stein and Wein [4] give a randomized $O(\log^2(m\mu)/\log\log(m\mu))$ approximation algorithm where μ is the maximum number of operations per job. This bound was slightly improved by Goldberg, Paterson, Srinivasan and Sweedyk [5]. Schmidt, Siegel and Srinivasan [6] give a deterministic $\log^2(m\mu)/\log\log(m\mu))$ approximation algorithm. When m is not fixed these are the best results known. For fixed m we have $(1+\epsilon)$ polynomial approximation schemes for all the three problems. An approximation scheme for flow shop was given by Hall [7]. Recently an approximation scheme for the open shop problem was given by Sevast'janov and Woeginger [8] while Solis-Oba and Sviridenko [9] give one for job shops.

Our contribution

In this paper we present a randomized approximation algorithm for flow shop scheduling when the number of machines is not fixed. Our algorithm is based on the rounding of the solution of an LP formulation of the flow shop problem. The LP formulation imposes some additional constraints which makes the rounding scheme possible. The makespan returned by our algorithm is within $2(1+\delta)$ times the optimal makespan and has an additive term of $O(m\ln(m+n)p_{\max})/\delta^2)$. This shows a tradeoff between the additive and multiplicative factors; the additive factor is better than the one in the Sevast'janov algorithm, and the multiplicative factor is better than that in the algorithm by Shmoys et al.

The remaining part of this paper is organized as follows. In Section 2, we discuss the new slotting constraints imposed by us. Section 3 gives a integral multicommodity flow formulation of the problem and Section 4 deals with the randomized algorithm and its analysis.

2 Slotting Constraints

It is no loss of generality to assume that all operations have size at least 1. Let $p_{\max}$ be the largest operation size. The machines are numbered in the order in which the operations of each job are to be processed.

We divide time into slots of size s, $s \geq 2p_{\max}$. For our randomized rounding scheme to work we require that the slots be *independent* of each other. By this we mean that the order in which operations are scheduled on a machine in any time-slot is independent of the order of the operations in other time-slots and on other machines. To ensure this we impose the restriction that no operation straddles a slot boundary and that no job moves from one machine to another in the middle of a slot. The second condition is equivalent to saying that the operations of a job are performed in distinct slots. Thus a job's operation on the i^{th} machine can only start if the operation on the $(i-1)^{th}$ machine has been completed by the end of the previous slot.

Consider a flow shop schedule with minimum makespan, OPT. We now show how to modify the schedule to satisfy the slotting constraints. First divide time into slots of size $s - p_{\max}$. Since there could be operations straddling slot-boundaries, we insert gaps of duration $p_{\max}$ after each slot. Operations starting in a slot and going over to the next now finish in these gaps. Finally we merge each gap with the slot just before it. This yields slots of size s and each operation now finishes in the slot in which it starts. Since OPT was the makespan of the original schedule the makespan of this modified schedule is $s \cdot OPT/(s - p_{\max})$. Next we shift all operations on the second machine by s, on the third machine by $2s, \ldots$, on the m^{th} machine by $(m-1)s$. This increases the makespan by $(m-1)s$ and gives a schedule which satisfies the restriction that all operations of a job are performed in distinct time-slots. Thus we have obtained a schedule which satisfies the slotting constraints and has makespan at most $\frac{s}{s-p_{\max}} OPT + (m-1)s$.

3 An integral multicommodity flow formulation

In this section we obtain an approximation to the flow shop scheduling problem with the slotting restriction. We begin by "guessing" the number of time-slots required by the schedule. Construct a directed graph $G = (V, E)$ which has a vertex for each (time-slot, machine) pair. There is an edge directed from vertex $u = (a, i)$ to vertex $v = (b, j)$ if $j = i + 1$ and $a < b$. For each job j we have two vertices s_j and t_j. s_j has edges to all vertices corresponding to the first machine and t_j has edges from all vertices corresponding to the last machine.

With this graph we associate a multicommodity flow instance; there is one commodity associated with each job j and s_j, t_j are the source and sink for this commodity. The flow of each commodity should be conserved: the total flow of a commodity entering a vertex (other than the source/sink of that commodity) is equal to the flow of that commodity leaving the vertex. We wish to route one unit of each commodity subject to the following throughput constraints on the vertices. Consider a vertex $v = (a, i)$ and let $x_{v,j}$ be the flow of commodity j through v. Then

$$\sum_j x_{v,j} p_{i,j} \leq s$$

Note that an integral multicommodity flow corresponds to a flow shop schedule satisfying the slotting restrictions. The feasibility of a multicommodity flow instance can be determined in polynomial time by formulating it as a linear program [10]. Infeasibilty of the multicommodity flow instance implies that our guess on the makespan is too small. Let k be the smallest number of time slots for which the multicommodity flow instance is feasible and let F be the corresponding flow. If T denotes the minimum makespan of a flow shop schedule satisfying the slotting constraints then $T \geq (k-1)s$.

4 The Algorithm and its Analysis

F is a (fractional) multicommodity flow which routes one unit of each commodity while respecting the throughput constraints on the vertices. Flow-theory says that the flow of any commodity can be viewed as a collection of at most $|E|$ paths. With each path we associate a weight which is just the flow along that path. Hence the total weight of all paths corresponding to commodity j is one.

The randomized algorithm picks exactly one path for each commodity with the probability of picking a path equal to its weight. This collection of paths, one for each commodity, gives an integral multicommodity flow which (possibly) violates the throughput constraints. The integral multicommodity flow in turn defines a flow shop schedule which satisfies the slotting constraints but which might be infeasible as the total processing time of operations schedule on a machine in a specific time-slot might exceed s.

Consider vertex $v = (a, i)$. Let X_j be a random variable which is 1 if job j is scheduled on machine i in slot a and 0 otherwise. Let X be a random variable defined as

$$X = \sum_j p_{ij} X_j$$

Claim. $E[X] \leq s$.

Proof. The probability that X_j is 1 equals $x_{v,j}$. Hence, $E[X_j] = x_{v,j}$. By linearity of expectations it follows that $E[X] = \sum_j p_{ij} E[X_j] = \sum_j p_{ij} x_{v,j}$. The throughput constraint on vertex v implies $\sum_j p_{ij} x_{v,j} \leq s$ from which the claim follows.

Claim. $\mathbf{Pr}[X > (1+\delta)s] \leq \left[\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right]^{s/p_{\max}}$.

Proof. Let X be the sum of n *independent Poisson* trials $X_1, X_2, \ldots, X_n$ with $\mathbf{Pr}[X_i = 1] = p_i$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^n p_i$. Using Chernoff bounds we obtain

$$\mathbf{Pr}[X > (1+\delta)\mu] < \left[\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right]^\mu$$

In our setting the Poisson trials $X_1, X_2, \ldots, X_n$ are such that $\mathbf{Pr}[X_j = 1] = x_{v,j}$ and $X = \sum_j p_{ij} X_j$. Consider the random variable $\tilde{X} = X/p_{\max}$. $\tilde{X}$ has mean $\tilde{\mu} = \mathbf{E}[X]/p_{\max}$. Let $s_j = p_{ij}/p_{\max}$; clearly $s_j \leq 1$.

For any $t > 0$ we have:

$$\mathbf{Pr}[X > (1+\delta)\mathbf{E}[X]] = \mathbf{Pr}[\tilde{X} > (1+\delta)\tilde{\mu}] = \mathbf{Pr}[e^{t\tilde{X}} > e^{t(1+\delta)\tilde{\mu}}]$$

By Markov's inequality it follows that

$$\mathbf{Pr}[e^{t\tilde{X}} > e^{t(1+\delta)\tilde{\mu}}] \leq \frac{\mathbf{E}\left[e^{t\tilde{X}}\right]}{e^{t(1+\delta)\tilde{\mu}}}.$$

Since $\tilde{X} = \sum_j s_j X_j$ and the random variables X_j are independent we obtain

$$\mathbf{E}\left[e^{t\tilde{X}}\right] = \mathbf{E}\left[\prod_j e^{ts_j X_j}\right] = \prod_j \mathbf{E}\left[e^{ts_j X_j}\right].$$

The random variable $e^{ts_j X_j}$ takes value e^{ts_j} with probability $x_{v,j}$ and the value 1 with probability $1 - x_{v,j}$. Hence,

$$\prod_j \mathbf{E}\left[e^{ts_j X_j}\right] = \prod_j [1 + x_{v,j}(e^{ts_j} - 1)]$$

Since $s_j \leq 1$, $e^{ts_j} - 1 \leq s_j(e^t - 1)$. Hence

$$\mathbf{E}\left[e^{t\tilde{X}}\right] \leq \prod_j [1 + x_{v,j} s_j (e^t - 1)] \leq \prod_j e^{x_{v,j} s_j (e^t - 1)}$$

where the last inequality uses the fact that for positive x, $e^x \geq 1 + x$. Now

$$\prod_j e^{x_{v,j} s_j (e^t - 1)} = e^{(e^t - 1) \sum_j x_{v,j} s_j} = e^{(e^t - 1)\tilde{\mu}}$$

Therefore,

$$\mathbf{Pr}[X > (1+\delta)s] \leq \mathbf{Pr}[X > (1+\delta)\mathbf{E}[X]] < \frac{e^{(e^t-1)\tilde{\mu}}}{e^{t(1+\delta)\tilde{\mu}}}.$$

Taking $t = \ln(1+\delta)$ gives us the best possible bound which is

$$\mathbf{Pr}[X > (1+\delta)s] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{s/p_{\max}}.$$

and this proves the claim

Observe that a trivial flow shop schedule satisfying the slotting constraints can be obtained by assigning job j to slot $j + i - 1$ on machine i, $1 \leq j \leq n$ and $1 \leq i \leq m$. This schedule has makespan $(n + m - 1)s$ and hence the number of vertices in the graph (excluding the source and sink vertices) is at most $m(n + m - 1)$.

Let $c = \frac{(1+\delta)^{(1+\delta)}}{e^{\delta}}$ and choose $s = \max\{2, \log_c[2m(n + m - 1)]\}p_{\max}$. From the above claim it follows that $\mathbf{Pr}[X > (1 + \delta)s] \leq \frac{1}{2m(n+m-1)}$. Hence the probability that for some machine and some time-slot the total processing time of operations scheduled on this machine in this time-slot is more than $(1 + \delta)s$ is at most 1/2. Equivalently, with probability at least 1/2 the processing time in every slot is less than $(1 + \delta)s$. Expanding each slot of size s to a slot of size $(1 + \delta)s$ then gives us a flow shop schedule of makespan $(1 + \delta)ks$.

Recall that any flow shop schedule satisfying the slotting restriction has makespan at least $(k-1)s$. Further, a flow shop schedule of makespan OPT yields a schedule satisfying the slotting restrictions and having makespan $\frac{s}{s-p_{\max}} OPT + (m-1)s$. This implies that

$$(k-1)s \leq \frac{s}{s - p_{\max}} OPT + (m-1)s$$

Hence with probability at least 1/2 the randomized rounding procedure gives us a feasible schedule whose makespan is bounded by

$$(1+\delta)ks \leq \frac{(1+\delta)s}{s - p_{\max}} OPT + (1+\delta)ms$$

where $s = \max\{2, \log_c[2m(n+m-1)]\}p_{\max}$, $c = \frac{(1+\delta)^{(1+\delta)}}{e^{\delta}}$ and δ is a positive constant chosen so that $c > 1$.

Theorem 4.1. *There exists a polynomial time randomized algorithm for flow shop scheduling which with probability at least 1/2 finds a schedule with makespan at most*

$$2(1+\delta)OPT + m(1+\delta)p_{\max}\log_c[2m(n+m-1)]$$

where $c = \frac{(1+\delta)^{(1+\delta)}}{e^{\delta}}$.

References

[1] S.M. Johnson. Optimal two- and three- stage procution schedules with setup times included. *Naval Research Logistics Quarterly*, 1:61–68, 1954.

[2] D.P. Williamson, L.A. Hall, J.A. Hoogeveen, C.A.J. Hurkens, J.K. Lenstra, and S.V. Sevast'janov. Short shop schedules. *Operations Research*, 45:288–294, 1997.

[3] S. Sevast'janov. Bounding algorithm for the routing problem with arbitrary paths and alternative servers. *Cybernetics*, 22:773–780, 1986.

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

[5] Leslie A. Goldberg, Mike Paterson, Aravind Srinivasan, and Elizabeth Sweedyk. Better approximation guarantees for job shop scheduling. In *Proc. 8th ACM-SIAM Symp. on Discrete Algorithms(SODA)*, pages 599–608, 1997.

[6] J.P. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-holding bounds for applications with limited independence. In *Proc. 4th ACM-SIAM Symp. on Discrete Algorithms(SODA)*, pages 331–340, 1993.

[7] Leslie A. Hall. Approximability of flow shop scheduling. In *Proc. 36th IEEE Annual Symp. on Foundations of Computer Science*, pages 82–91, 1995.

[8] S. Sevast'janov and G. Woeginger. Makespan minimization in open shops : A polynomial time approximation scheme. *Mathematical Programming*, 82(1-2), 1998.

[9] Maxim Sviridenko, K. Jansen, and R. Solis-Oba. Makespan minimization in job shops : A polynomial time approxmiation scheme. In *Proc. 31st Annual ACM Symp. on Theory of Computing*, pages 394–399, 1999.

[10] D.B. Shmoys. *Approximation algorithms for* **NP**-*Hard problems*, chapter Cut problems and their applications to divide and conquer, pages 192–234. PWS, 1997.