Characterizing the Distribution of Low-Makespan Schedules in the Job Shop Scheduling Problem

Matthew J. Streeter[1] **and Stephen F. Smith**[2]
Computer Science Department
and Center for the Neural Basis of Cognition[1] and
The Robotics Institute[2]
Carnegie Mellon University
Pittsburgh, PA 15213
{matts, sfs}@cs.cmu.edu

Abstract

We characterize the search landscape of the job shop scheduling problem (JSSP), with a focus on schedules whose makespan is optimal or near-optimal. Building on previous work on the 'big valley' distribution of local optima, we use special branch and bound algorithms to examine in greater detail the extent to which JSSP search spaces conform to the intuitive picture conveyed by the words 'big valley'. We also examine how this changes as a function of the job:machine ratio. We find that for square JSSPs, low-makespan schedules are tightly clustered in a small region of the search space, and the size of this region decreases as the makespan gets closer to optimality. As the job:machine ratio increases beyond 1, however, low-makespan schedules become dispersed throughout the search space. We discuss the reasons for this and provide analytical results for two limiting cases. We close with an examination of neighborhood exactness in the JSSP, which illustrates some limitations of the big valley picture for JSSP landscapes.

1. Introduction

1.1. Motivations

Local search algorithms (e.g., taboo search (Glover & Laguna 1997; Nowicki & Smutnicki 1996), iterated local search (Lourenço, Martin, & Stützle 2003)) are among the most effective approaches for solving the job shop scheduling problem (Jain & Meeran 1998; Jones & Rabelo 1998). Such algorithms traverse the search space from point to point, employing a bias designed to move them toward optimal or near-optimal schedules. Clearly, the algorithms that have proven the most effective are those whose bias is in some sense well-matched to the unique and special structure of the JSSP. In some cases, algorithm design decisions have been explicitly motivated by observed search space properties. For example, Nowicki and Smutnicki motivate the use of *path relinking* in their *i*-TSAB algorithm by citing evidence that the JSSP has a 'big valley' distribution of local optima (Nowicki & Smutnicki 2001; 2002). The picture of the search space given in this paper is much richer than that provided by studies of the 'big valley' distribution, and we hope that it too will prove valuable in guiding the design of search algorithms for the JSSP.

In this paper, we will focus on characterizing how *optimal* or *near-optimal* schedules are distributed in random instances of the JSSP. Though near-optimal schedules represent only a vanishingly small portion of the search space, our focus on them makes sense because these are the schedules one wants to find. We do expect such a characterization to be prescriptive with regard to search algorithm design. For example, if near-optimal schedules are found in a single cluster whose centroid is close to the centroid of globally optimal schedules, one might expect that starting a local search algorithm from a schedule that lies along the path between two near-optimal schedules (as is done in *i*-TSAB) would generate better results than would be obtained using a random starting point. On the other hand, if near-optimal schedules are dispersed throughout the search space and their centroid is not particularly close to any global optimum, this might be less effective.

1.2. Contributions

The contributions of this paper are twofold. First, we introduce techniques that could be usefully applied to the analysis of other scheduling problems such as permutation flow shop or project scheduling, as well as other combinatorial problems. Second, we make use of these techniques to obtain insights into the job shop scheduling problem.

With regard to techniques, we present two ways to use branch and bound algorithms to compute search space statistics that otherwise could not be computed tractably. First, we show how to calculate the number of attributes common to all schedules whose makespan does not exceed a given threshold by using multiple runs of branch and bound. Second, we use branch and bound to efficiently find the best schedule within a fixed hamming radius of a given center schedule, and show how this can be used to quantitatively measure the 'ruggedness' of the landscape.

In terms of insights into the job shop scheduling problem, the most significant results of our analysis are listed below.

- Empirically, we demonstrate that for low job:machine ratios, low-makespan schedules are clustered in a small region of the search space and many attributes (i.e., directed disjunctive graph edges) are common to all low-makespan schedules. As the job:machine ratio increases, low-makespan schedules become dispersed throughout the search space and there are no attributes common to all low-makespan schedules.

- For two limiting cases we derive analytical results. We prove that as the job:machine ratio approaches 0, all globally optimal schedules will have a common orientation of a randomly selected disjunctive edge with probability approaching 1. As the job:machine ratio approaches ∞ this happens with probability approaching 0.
- We present the first formalization of the concept of a 'big valley' distribution, give techniques for quantitatively measuring the degree to which a search landscape has such a distribution, and apply these techniques to the JSSP. We find that the search landscape for random square JSSPs can indeed be thought of as a single 'big valley', but that this requires taking a very coarse-grained view of the landscape.

Our work is suggestive with regard to understanding average-case difficulty of random JSSP instances as a function of their job:machine ratio, particularly in light of the conventional wisdom that square JSSPs are more difficult than rectangular ones (Fisher & Thompson 1963). For the purposes of this paper, however, we only attempt to describe the search space, and do not relate this description to instance difficulty seen by particular algorithms.

2. Related Work

There are at least three threads of research that have conducted search space analyses related to the ones we conduct here. These include literature on the 'big valley' distribution common to a number of combinatorial optimization problems, studies of backbone size in boolean satisfiability, and a statistical mechanical analysis of the TSP.

2.1. The Big Valley

The term 'big valley' originated in a paper by Boese et al. (1994) that examined the distribution of local optima in the Traveling Salesman Problem (TSP). Based on a sample of local optima obtained by next-descent starting from random TSP tours, Boese calculated two correlations:

1. the correlation between the cost of a locally optimal tour and its average distance to other locally optimal tours, and
2. the correlation between the cost of a locally optimal tour and the distance from that tour to the best tour in the sample.

The distance between two TSP tours was defined as the total number of edges minus the number of edges that are common to the two tours. Based on the fact that both of these correlations were surprisingly high, and the fact that the mean distance between random local optima was small relative to the mean distance between random tours, Boese conjectured that local optima in the TSP are arranged in a 'big valley'. The term does not have a formal definition, but intuitively it means that if the search space could be locally smoothed in some manner it would then have a single basin of attraction with respect to common TSP move operators such as Lin 2-opt.

Boese's analysis has been applied to other combinatorial problems (Kim & Moon 2004), including the permutation flow shop scheduling problem (Watson *et al.* 2002; Reeves & Yamada 1998) and the job shop scheduling problem (Nowicki & Smutnicki 2001). Correlations observed for the job shop problem are generally weaker than those observed for the TSP.

2.2. Backbone Size

The *backbone* of a problem instance is the set of attributes common to all globally optimal solutions of that instance. For example, in the boolean satisfiability problem (SAT), the backbone is the set of variable assignments that are common to all satisfying assignments. In the JSSP, the backbone could be defined as the number of disjunctive edges (described in §3.1) that have a common orientation in all globally optimal schedules. Backbone size is relevant to our work in that it is the leftmost data point on each of the graphs in figures 1 and 2.

There is a large literature on backbones in combinatorial optimization problems, including many empirical and analytical results (Slaney & Walsh 2001; Monasson *et al.* 1999). In an analysis of problem difficulty in the JSSP, Watson et al. (2001) present histograms of backbone size for random 6x6 (6 job, 6 machine) and 6x4 (6 job, 4 machine) JSSP instances. Summarizing experiments not reported in their paper, Watson et al. note that "[F]or [job:machine ratios] $>$ 1.5, the bias toward small backbones becomes more pronounced, while for ratios $<$ 1, the bias toward larger backbones is further magnified." §4 generalizes these observations and proves two theorems that gives insight into why this phenomenon occurs.

2.3. Statistical Mechanical Analyses

A large and growing literature applies techniques from statistical mechanics to the analysis of combinatorial optimization problems (Martin, Monasson, & Zecchina 2001). At least one result obtained in this literature concerns clustering of low-cost solutions. In a study of the TSP, Mézard and Parisi (1986) obtain an expression for the expected overlap (number of common edges) between random TSP tours drawn from a Boltzmann distribution. They show that as the temperature parameter of the Boltzmann distribution is lowered (placing more probability mass on low-cost TSP tours), expected overlap approaches 100%. Though we do not use a Boltzmann weighting, §5 of this paper examines how expected overlap between random JSSP schedules changes as more probability mass is placed on low-makespan schedules.

3. The Job Shop Scheduling Problem

3.1. Problem Definition

An instance of the job shop scheduling problem consists of N jobs and M machines, where each of the jobs is a sequence of M operations (exactly one per machine) that must be performed in a certain order. Each operation o has an associated duration $\tau(o)$. A schedule assigns start times to each operation such that

1. no machine is scheduled to process more than one operation at the same time, and
2. the ordering of operations in each job is respected.

The *makespan* of a schedule is equal to the maximum completion time (i.e., start time plus duration) of any operation. We consider the makespan-minimization version of the job shop scheduling problem, in which the objective is to find a schedule that minimizes the makespan.

It is convenient to represent a schedule in terms of its *disjunctive graph* (Roy & Sussmann 1964). In a disjunctive graph, there is a vertex corresponding to each operation in the problem instance, as well as two special vertices called the *source* and the *sink*. There are directed edges pointing from the source into (the vertex corresponding to) the first operation of each job, from the last operation of each job into the sink, and from each operation into the next operation (if any) in the job. A directed edge from operation o_1 to operation o_2 indicates that o_1 completes before o_2 starts. The orientation of all the edges just discussed is dictated by the problem instance, and these edges are called *conjunctive edges*. The remaining edges connect fixed pairs of operations, but their orientation is defined by a particular schedule. These *disjunctive edges* connect all pairs of operations performed on the same machine. Once the orientation of all edges is specified, the weight of each edge is given by the duration of the operation that the edge points out of (or zero if the edge points out of the source). It can be shown that the makespan of the schedule is equal to the length of the longest weighted path from the source to the sink.

In this way, the job shop scheduling problem can be rephrased as a task of specifying a binary orientation for each of $M\binom{N}{2}$ disjunctive edges.

3.2. Distance between schedules

We define the distance between two schedules as the number of disjunctive edges that are oriented in opposite directions in the two schedules. Unless otherwise noted we will normalize the distance by dividing by the total number of disjunctive edges (Mattfeld, Bierwirth, & Kopfer 1999).

3.3. Random Instances

Definition (Random JSSP instance). *A random N by M instance of the JSSP is an instance with N jobs and M machines, where the order in which the machines appear in each job is a random permutation of $\{1, 2, ..., M\}$ and where the durations of each operation are drawn independently from a distribution with finite mean μ and finite variance σ^2.*

We choose operation durations from a uniform distribution over $\{1, 2, ..., 100\}$ for the experiments described in this paper.

4. Number of Common Attributes as a Function of Makespan

In this section we consider schedules whose makespan is within a certain factor ρ of optimality, and look at the fraction of disjunctive edges that will have the same orientation in all such schedules. We will make use of the following definition (a related definition is used in (Slaney & Walsh 2001)).

Definition (ρ-backbone). *The ρ-backbone of a JSSP instance is the set of disjunctive edges that have a common orientation in all schedules whose makespan is at most ρ times the optimal makespan.*

In this section we examine the expected value of $|\rho$-backbone$|$ as a function of ρ for random *N* by *M* JSSP instances. We specifically examine how the shape of this curve changes as we change $\frac{N}{M}$. To do this we must compute $|\rho$-backbone$|$ for all ρ for an arbitrary JSSP instance. We do this by performing a separate run of branch and bound associated with each possible orientation of each disjunctive edge.

4.1. Methodology

Let $opt(o_1, o_2)$ denote the optimal makespan among all schedules in which operation o_1 is performed before operation o_2, where o_1 and o_2 are operations performed on the same machine. In branch and bound algorithms for the JSSP, nodes in the search tree represent choices of orientations for a subset of the disjunctive edges. Thus, by constructing a root search tree node that has an edge pointing from o_1 into o_2, we can determine $opt(o_1, o_2)$ using existing branch and bound algorithms. We use an algorithm due to Brucker et al. (1994) because it is efficient and because the code for it is freely available via ORSEP.

Suppose we are interested in all schedules whose makespan is at most $t = \rho \cdot opt\_makespan$, where $opt\_makespan$ is the optimum makespan for the given instance. Consider a pair of operations o_1 and o_2, performed on the same machine. It must be the case that $\min(opt(o_1, o_2), opt(o_2, o_1)) = opt\_makespan$, because any globally optimal schedule must either perform o_1 before o_2 or perform o_2 before o_1. If $t < \max(opt(o_1, o_2), opt(o_2, o_1))$, then the presence of one of these two directed edges precludes schedules with makespan $\leq t$, so the other edge must be a member of the ρ-backbone. Thus $|\rho$-backbone$|$ is equal to

$$\sum_{o_1 \neq o_2, m(o_1)=m(o_2)} \frac{1}{2}[t < \max(opt(o_1, o_2), opt(o_2, o_1))]$$

where $m(o)$ is the machine on which an operation o is performed, and the [...] notation indicates a function that returns 1 if the predicate enclosed in the brackets is true, 0 otherwise.

Thus, we can determine $|\rho$-backbone$|$ for all ρ by performing enough branch and bound runs to determine $opt(o_1, o_2)$ for all pairs of distinct operations performed on the same machine.

These values can be determined using $1 + M\binom{N}{2}$ runs of branch and bound. The first branch and bound run is used to find a globally optimal schedule, which gives the value of opt for one of the two possible orientations of each of the $M\binom{N}{2}$ disjunctive edges. A separate branch and bound

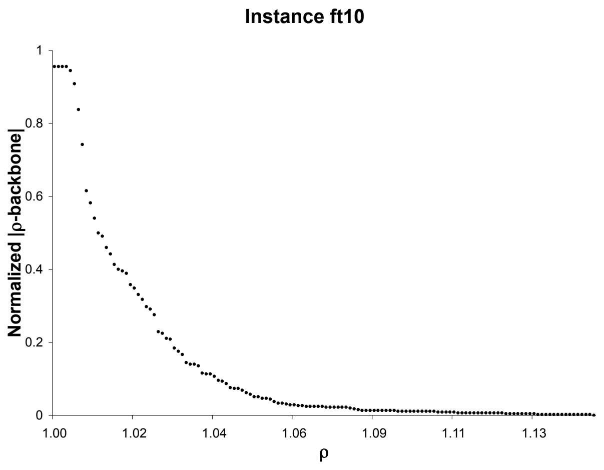

Figure 1: Normalized $|\rho$-backbone$|$ as a function of ρ for OR library instance ft10.

run is used to determine the values of opt for the $M\binom{N}{2}$ alternative orientations.

Figure 1 graphs the fraction of disjunctive edges that belong to the ρ-backbone as a function of ρ for instance ft10 from the OR library (Beasley 1990). Note that by definition the curve is non-increasing with respect to ρ, and that the curve is exact for all ρ. It is noteworthy that among schedules within 0.5% of optimality, 80% of the disjunctive edges have a fixed orientation. We will see that this behavior is typical of square JSSP instances.

4.2. Experiments on Random Instances

We plotted $|\rho$-backbone$|$ as a function of ρ for all instances in the OR library having 10 or fewer jobs and 10 or fewer machines. The results are available online (Streeter & Smith 2005). Inspection of the graphs revealed that the shape of the curve is largely a function of the job:machine ratio. To investigate this further, we repeated these experiments on a large number of randomly generated JSSP instances.

We use randomly-generated instances with 7 different combinations of N and M to study instances with $\frac{N}{M}$ equal to 1, 2, or 3. For $\frac{N}{M} = 1$ we use 6x6, 7x7, and 8x8 instances; for $\frac{N}{M} = 2$ we use 8x4 and 10x5 instances; and for $\frac{N}{M} = 3$ we use 9x3 and 12x4 instances. We generate 1000 random instances for each combination of N and M.

Figure 2 presents the expected fraction of edges belonging to the ρ-backbone as a function of ρ for each combination of N and M, grouped according to $\frac{N}{M}$. For the purposes of this study the two most important observations about Figure 2 are as follows.

- The curves depend on both the size of the instance (i.e., NM) and the shape (i.e., $\frac{N}{M}$). Of these two factors, $\frac{N}{M}$ has by far the stronger influence on the shape of the curves.
- For *all* values of ρ, the expected fraction of edges belonging to the $|\rho$-backbone$|$ decreases as $\frac{N}{M}$ increases.

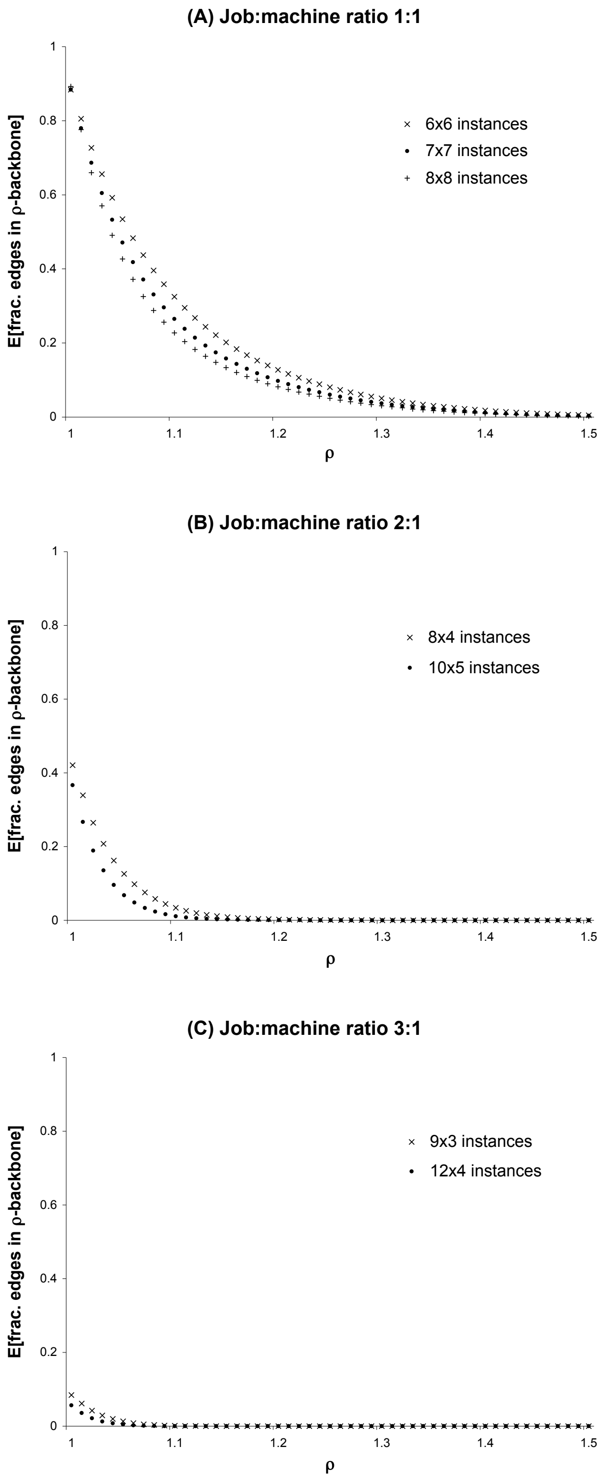

Figure 2: Expected fraction of edges in ρ-backbone as a function of ρ for random JSSP instances. Graphs (A), (B), and (C) depict curves for random instances with $\frac{N}{M}$ = 1, 2, and 3, respectively.

4.3. Analysis

We now give some insight into Figure 2 by analyzing two limiting cases. We prove that as $\frac{N}{M}\rightarrow 0$, the expected fraction of disjunctive edges that belong to the backbone approaches 1, while as $\frac{N}{M}\rightarrow\infty$ this expected fraction approaches 0.

Intuitively, what happens is as follows. As $\frac{N}{M}\rightarrow 0$ (i.e., N is held constant and $M\rightarrow\infty$) each of the jobs becomes very long. Individual disjunctive edges then represent precedence relations among operations that should be performed very far apart in time. For example, if there are 10,000 machines (and so each job consists of 10,000 operations), a disjunctive edge might specify whether operation 1,200 of job A is to be performed before operation 8,500 of job B. Clearly, waiting for job B to complete 8,500 of its operations before allowing job A to complete 12% of its operations is likely to produce an inefficient schedule. Thus, orienting a single disjunctive edge in the 'wrong' direction is likely to prevent a schedule from being optimal, and so any particular edge will likely have a common orientation in all globally optimal schedules.

In contrast, when $\frac{N}{M}\rightarrow\infty$, it is the workloads of the machines that become very long. The order in which the jobs are processed on a particular machine does not matter much as long as the machine with the longest workload is kept busy, and so the fact that a particular edge is oriented a particular way cannot prevent a schedule from being optimal. All of this is formalized below.

We will make use of the following well-known definition and inequalities.

Definition (w.h.p). *A sequence of events ξ_n occurs with high probability (w.h.p) if* $\lim_{n\rightarrow\infty}\mathbb{P}[\xi_n]=1$.

Bonferroni's Inequality. $\mathbb{P}[\bigcup_{i=1}^{n}E_i]\leq\sum_{i=1}^{n}\mathbb{P}[E_i]$, *where E_1, E_2, ..., E_n are events, and $\mathbb{P}[\bigcup_{i=1}^{n}E_i]$ is the probability that at least one of these events occurs.*

Chebyshev's Inequality. $\mathbb{P}[|X-\mu|\geq k\sigma]\leq\frac{1}{k^2}$, *where $k\geq 0$ and X is a random variable drawn from a distribution with mean μ and variance σ^2.*

Lemma 1 and Theorem 1 show that for constant N, a randomly chosen edge of a random N by M JSSP instance will be in the backbone w.h.p (as $M\rightarrow\infty$). Lemma 2 and Theorem 2 show that for constant M, a randomly chosen edge of a random N by M JSSP instance will not be in the backbone w.h.p (as $N\rightarrow\infty$). Because the proof of Lemma 2 is somewhat long and technical, we have moved in to Appendix A.

Lemma 1. *Let $I_{N,M}$ be a random N by M JSSP instance. Let f be an unbounded, increasing function of M. Then w.h.p (as $M\rightarrow\infty$) the optimum makespan of $I_{N,M}$ does not exceed $M\mu+\sigma\sqrt{M}f(M)$.*

Proof. Let D_J be the sum of the durations of all operations in job J. From the central limit theorem, each D_J has mean $M\mu$ and standard deviation $\sigma\sqrt{M}$. The inequality $\max_{1\leq J\leq N}D_J\leq M\mu+\sigma\sqrt{M}(f(M)-1)$ holds w.h.p (as can be shown using Chebyshev's inequality). Thus it suffices to show that w.h.p a schedule exists whose makespan does not exceed $(\max_{1\leq J\leq N}D_J)+\sigma\sqrt{M}$.

Let $H(o)$ be the sum of the durations of all operations that come before o in the job that o is part of. Assume without loss generality that all $H(o)$ are distinct. Let the (infeasible) schedule H assign start time $H(o)$ to each operation o. The makespan of H is $\max_{1\leq J\leq N}D_J$. Let an operation o be *running* at time t under schedule S (which assigns start times $S(o)$) if $S(o)\leq t\leq S(o)+\tau(o)$ (recall that $\tau(o)$ is o's duration). Let an operation $\bar{o}$ be *infeasible* under S if any operation performed on the same machine as $\bar{o}$ is running at time $S(\bar{o})$ under S. Let $\bar{O}(S)$ be the set of operations that are infeasible under S.

It suffices to show that we can transform H into a feasible schedule while increasing its makespan by an amount whose expected value does not increase with M. We do this using an operator called $shift$. For any schedule S, time t, and increment Δ, let $shift(S,t,\Delta)$ be a schedule identical to S, except that all operations whose start time was $\geq t$ have had their start time incremented by Δ. Applying $shift$ cannot create any new infeasible operations. Furthermore, for any $\bar{o}\in\bar{O}(H)$, applying $shift(H,H(\bar{o}),\Delta(\bar{o}))$ for sufficiently large $\Delta(\bar{o})$ creates a schedule in which $\bar{o}$ is no longer infeasible. It is sufficient to set $\Delta(\bar{o})=(\max_{o\neq\bar{o},o\text{ running at }H(\bar{o})}H(o)+\tau(o))-H(\bar{o})$. Iterating this idea yields a feasible schedule $\hat{H}$ with start times given by

$$\hat{H}(o)\equiv H(o)+\sum_{\bar{o}\text{ infeasible under }H;\ H(\bar{o})\leq H(o)}\Delta(\bar{o})\ .$$

Consider the difference between the makespan of $\hat{H}$ and that of H. Under H, there are at most N concurrent operations at any given time, each one equally likely to be performed on any of the M machines. Thus the probability that a random operation o is infeasible under H cannot exceed $\frac{N-1}{M}$ by Bonferroni's inequality. Thus the expected number of infeasible operations cannot exceed $NM\frac{N-1}{M}=N(N-1)$. Thus $\mathbb{E}[makespan(\hat{H})-makespan(H)]$ cannot exceed $\mathbb{E}[\Delta]N(N-1)$. Using the fact that σ is finite it can be shown that $\mathbb{E}[\Delta]$ does not increase with M. Thus $\mathbb{E}[makespan(\hat{H})-makespan(H)]$ does not increase with M, as required.

□

Theorem 1. *Let N be constant, let $I_{N,M}$ be a random N by M JSSP instance, and let e be a randomly selected disjunctive edge of $I_{N,M}$. Then w.h.p (as $M\rightarrow\infty$) e is in the backbone of $I_{N,M}$.*

Proof. We show that w.h.p one of the two possible orientations of e will create a path through the disjunctive graph with length in excess of $M\mu+\sigma\sqrt{M}log(M)$. By Lemma 1, this implies that e is in the backbone of $I_{N,M}$ w.h.p.

Let e be between vertices corresponding to the i^{th} operation of one job to the j^{th} operation of another, and consider orienting e so that the operation whose position in its job is $\max(i,j)$ is performed before the operation whose position is $\min(i,j)$. This orientation creates a path P with weighted length $w(P)$ that passes through $|P|=M+1+\max(i,j)-\min(i,j)$ operations. $|P|$ exceeds $M+M^{\frac{2}{3}}$ w.h.p (because i is not within $M^{\frac{2}{3}}$ operations of j w.h.p).

If $|P| > M + M^{\frac{2}{3}}$, $w(P)$ exceeds $M\mu + \sigma\sqrt{M}log(M)$ w.h.p. To see this, consider some fixed P, and note that by the central limit theorem $w(P)$ has mean $\mu\,|P|$ and standard deviation $\sigma\sqrt{|P|} \le \sigma\sqrt{2M}$. $w(P)$ will be within $\log(M)$ standard deviations of its mean w.h.p, and will therefore exceed $(M + M^{\frac{2}{3}})\mu - (\sigma\sqrt{2M})\log(M)$ w.h.p. This quantity exceeds $M\mu + \sigma\sqrt{M}\log(M)$ for M sufficiently large.

□

Lemma 2. *Let M be constant and let $I_{N,M}$ be a random N by M JSSP instance. Then w.h.p (as $N\to\infty$) there exists an optimal schedule for $I_{N,M}$ with the property that no machine is idle until all operations performed on that machine have been completed.*

Proof. See appendix A. □

Corollary 1. *Let M be constant and let $I_{N,M}$ be a random N by M JSSP instance. Let W_m denote the sum of the durations of all operations performed on machine m. Then w.h.p (as $N\to\infty$) the optimum makespan of $I_{N,M}$ is equal to $\max_{1\le m\le M} W_m$.*

Corollary 1 confirms a conjecture of Taillard (1994).

Theorem 2. *Let M be constant, let $I_{N,M}$ be a random N by M JSSP instance, and let e be a randomly selected disjunctive edge of $I_{N,M}$. Then w.h.p (as $N\to\infty$) e is not in the backbone of $I_{N,M}$.*

Proof. Let e specify that operation o_1 is to be performed before operation o_2, where o_1 and o_2 are operations performed on the same machine as part of jobs j_1 and j_2, respectively. Remove j_2 from $I_{N,M}$ to create a new instance $I_{N-1,M}$, which comes from exactly the same distribution as a fresh random $N-1$ by M instance. Lemma 2 shows that w.h.p there exists a schedule for $I_{N-1,M}$, call it S, such that no machine is idle until all its operations have finished. It suffices to show that w.h.p we can work j_2 into S to create a schedule that is optimal for $I_{N,M}$ and that performs o_1 before o_2.

Let W_m denote the workload of machine m (i.e., the sum of the durations of all operations performed on m) in $I_{N-1,M}$. Let m_i be the machine with the i^{th} highest workload. Let $\bar{o}$ be the one operation of j_2 that uses m_1. The optimum makespan of $I_{N,M}$ is at least $W_{m_1} + \tau(\bar{o})$. Let L be the set of operations with start time $\ge W_{m_2}$ under S. Let $w(O)$ denote the sum of the durations of an arbitrary set of operations O. All operations in L must be performed on m_1 and must be the last operations of their jobs, so we can permute the order of operations in L arbitrarily without changing the makespan or feasibility of S. If $o_1 \in L$ we assume without loss of generality that $S(o_1) = W_{m_2}$. It then suffices to show that w.h.p the operations in $j_2 \cup (L \setminus o_1)$ can be scheduled in a window of length $w(L \setminus o_1) + \tau(\bar{o})$.

It is easy to see that this can always be done if $(L \setminus o_1)$ can be partitioned into two subsets, L_1 and L_2, such that $w(j_2) \le w(L_1) \le w(L_2)$, where $w(j_2)$ is the sum of the durations of the operations in job j_2. Because $\mathbb{E}[w(j_2)] = M\mu$ (a constant), it can be shown that if $w(L) > \log(N)$ such a partitioning exists w.h.p. We have $w(L) = W_{m_1} - W_{m_2} \ge \min_{i\ne j}\left|W_{m_i} - W_{m_j}\right|$. Asymptotically, each W_{m_i} is normally distributed with mean $N\mu$ and variance $N\sigma^2$ (by the central limit theorem). Thus we can have $\min_{i\ne j}\left|W_{m_i} - W_{m_j}\right| \le \log(N)$ only if two among these M normally distributed random variables are within $\frac{\log(N)}{\sigma\sqrt{N}}$ standard deviations of one another. But this does not happen w.h.p.

□

Note that the expected fraction of edges in the backbone is equal to the probability that a random edge is part of the backbone by linearity of expectations.

5. Clustering as a Function of Makespan

§4 examined the number of attributes common to all schedules whose makespan is within a factor ρ of optimality. In this section, we examine the *expected distance* between random members of the set of such schedules. Note that the number of attributes common to a set of schedules provides an upper bound on the expected distance between random schedules in the set (because common attributes can never contribute to distance). However, it provides no lower bound. Thus, even though global optima of random instances with $\frac{N}{M} = 3$ do not share many attributes, they may still be tightly clustered in the search space. By computing the expected distance between random schedules drawn from among the set of schedules within a factor ρ of optimality, we can determine whether or not this is the case.

5.1. Methodology

We generate ‘random’ samples from the set of schedules within a factor ρ of optimality by running the simulated annealing algorithm of van Laarhoven *et al.* (1992) until it finds such a schedule. More precisely, our procedure for sampling distances is as follows.

1. Generate a random N by M JSSP instance I.
2. Using the branch and bound algorithm of Brucker et al. (1994), determine the optimal makespan of I.
3. Perform two runs, R_1 and R_2, of the van Laarhoven et al. (1992) simulated annealing algorithm. Restart each run as many times as necessary for it to find a schedule whose makespan is optimal.
4. For each ρ in $\{1, 1.01, 1.02, ..., 2\}$, find the first schedule, call it $s_i(\rho)$, in each run R_i whose makespan is within a factor ρ of optimality. Add the distance between $s_1(\rho)$ and $s_2(\rho)$ to the sample of distances associated with ρ.

Note that the distances for different values of ρ are dependent, but that for a given ρ all the sampled distances are independent, which is all that is necessary for the estimates to be unbiased. Figure 3 presents the results of running this procedure on 1000 random JSSP instances for the same 7 combinations of N and M that were used in §4.2.

From examination of Figure 3, we see that for ρ near 1, the ρ-optimal schedules are in fact dispersed widely throughout the search space for $\frac{N}{M} = 3$, and that this is true to a lesser extent for $\frac{N}{M} = 2$.

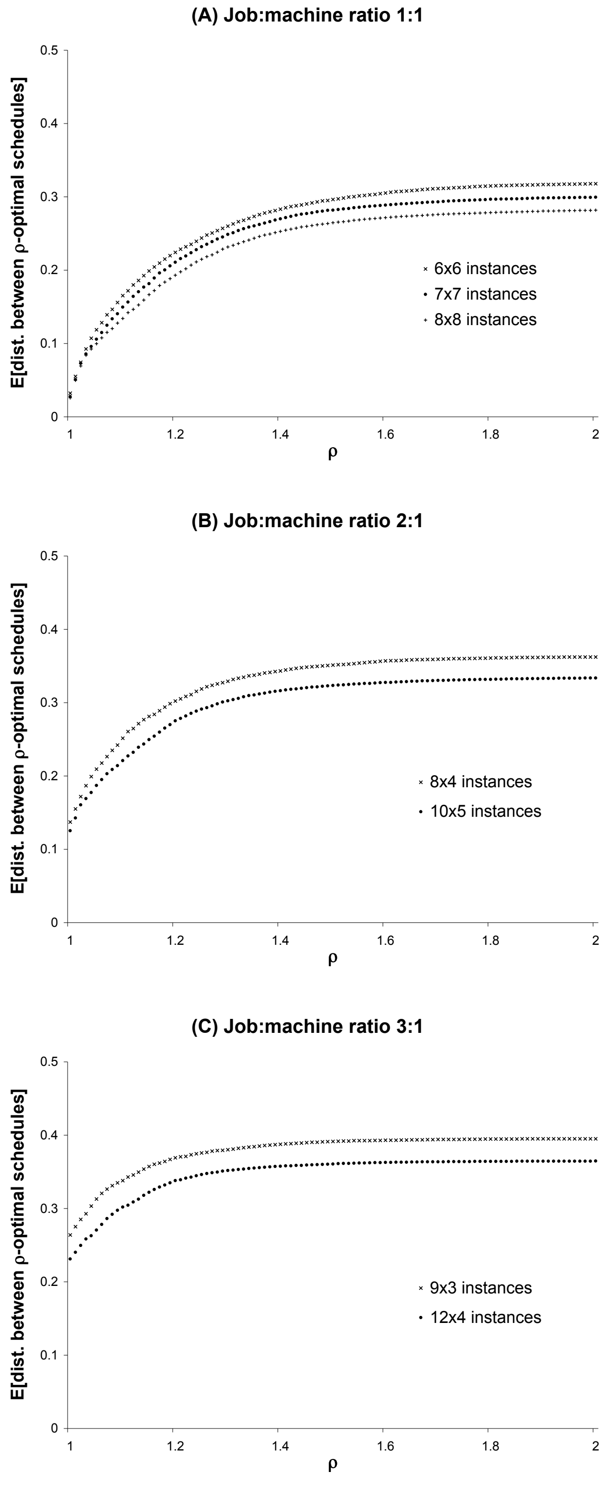

Figure 3: Expected distance between random schedules within a factor ρ of optimality, as a function of ρ. Graphs (A), (B), and (C) depict curves for random instances with $\frac{N}{M}$ = 1, 2, and 3, respectively.

An immediate implication of Figure 3 is that whether or not they exhibit the two correlations that are the operational definition of a 'big valley', typical landscapes for JSSP instances with $\frac{N}{M} = 3$ cannot be expected to be 'big valleys' in the intuitive sense of these words. If anything, one might posit the existence of multiple 'big valleys', each leading to a separate global optimum. We make use of these observations in the discussion in §6.4.

6. Neighborhood Exactness

6.1. Motivations

Intuitively, a search landscape can be described by the words 'big valley' if, when sufficiently smoothed in some manner, the search landscape has a single basin of attraction. In this section, we consider a type of smoothing in which an algorithm performing next-descent is allowed to jump to any improving schedule within a distance r of its current location. We accomplish this smoothing using special branch and bound algorithms that find the best schedule within a radius r of an arbitrary schedule s. Intuitively, this smoothing allows next-descent to jump over local irregularities in the search landscape and explore the landscape more globally. We examine the probability that a random local optimum (i.e., a local optimum obtained by applying next-descent to a random schedule) of a random N by M JSSP instance will be globally optimal, as a function of r. The resulting curve is a quantitative measure of the degree to which typical landscapes of random N by M JSSP instances can be described as consisting of one or more 'big valleys'.

6.2. Methodology

To formalize what we are doing it is helpful to introduce the following two definitions.

Definition (Neighborhood $\mathcal{N}_r$). *Let $\mathcal{N}_r(s)$ denote the set of all schedules whose disjunctive graph distance from s (i.e., number of disjunctive edges whose orientation differs from that in s) is at most r.*

Definition (Neighborhood exactness). *Let S be a set of points in a search space, let $f : S \rightarrow \Re$ be an objective function to be minimized, and let $\mathcal{N} : S \rightarrow 2^S$ be a neighborhood function. Let $\mathcal{P}$ be a probability distribution over points in S that are locally optimal w.r.t. $\mathcal{N}$. The exactness of the triple $(\mathcal{N}, \mathcal{P}, S)$ is the probability that a random local optima drawn from $\mathcal{P}$ is globally optimal with respect to f.*

Note that the exactness of a neighborhood is 1 if the neighborhood is exact under the standard definition of this term (namely, that all local optima are global optima).

For given values of N and M, we compute *expected* exactness of $\mathcal{N}_r$ for $1 \leq r \leq M\binom{N}{2}$ by repeatedly executing the following procedure (which implicitly specifies $\mathcal{P}$).

1. Generate a random N by M JSSP instance I.
2. Using the algorithm of Brucker et al. (1994), compute the optimal makespan of I.
3. Let s be a random feasible schedule; let $r = 1$; and let $opt = false$.
4. While $opt = false$ do:

(a) Starting from s, apply next-descent under the neighborhood N_r to generate a local optimum (each step of next-descent uses our radius-limited branch and bound algorithm). Let s be this local optimum.

(b) Update the expected exactness of $\mathcal{N}_r$ appropriately, based on whether or not s is globally optimal.

(c) If s is globally optimal, set $opt = true$. Otherwise increment r.

5. For all r' such that $r < r' \leq M\binom{N}{2}$ update the expected exactness of $\mathcal{N}_{r'}$ appropriately.

For distinct radii, the estimates of expected exactness given by this procedure are dependent. However, all data that contribute to the estimate for a given radius are independent. The fact that r increases gradually does not change the fact that we are doing next-descent w.r.t each individual r.

Our radius-limited branch and bound algorithm uses the branching rule of Balas (1969) combined with the lower bounds and branch ordering heuristic of Brucker et al. (1994).

6.3. Results

For this experiment, we use different combinations of N and M than were used in §4 and §5. We use combinations with $\frac{N}{M} = \frac{1}{5}$ (3x15, 4x20, and 5x25 instances), $\frac{N}{M} = 1$ (6x6, 7x7, and 8x8 instances) and $\frac{N}{M} = 5$ (15x3 and 20x4 instances). As before, we generate 1000 random instances for each combination of N and M. We use different combinations of N and M in this section because the trends in this data are only apparent when using combinations with $\frac{N}{M} < 1$ as well as with $\frac{N}{M} > 1$, and because we found that, in these experiments, more extreme job:machine ratios were necessary to bring the trends out clearly.

Figure 4 plots expected exactness as a function of neighborhood radius (normalized by the number of disjunctive edges) for each of these three job:machine ratios.

6.4. Discussion

When $\frac{N}{M} = \frac{1}{5}$ or $\frac{N}{M} = 5$, only a small amount of smoothing is required before a random local optimum drawn from a random instance is very likely to be globally optimal. Using the methodology of §4, we found that the expected backbone fractions for 3x15, 4x20, and 5x25 instances are 0.94, 0.93, and 0.92, respectively, while the expected distance between global optima was 0.02 in all three cases. This suggests that the typical landscape for $\frac{N}{M} = \frac{1}{5}$ can be described as a 'big valley'. In contrast, the expected backbone fractions for 15x3 and 20x4 instances are near-zero, while the expected distances between global optima are 0.33 and 0.28, respectively. Thus for $\frac{N}{M} = 5$, the data suggest the existence of many 'big valleys' rather than just one.

For $\frac{N}{M} = 1$, normalized radius r must be much larger in order to achieve the same expected exactness. The data from §5 show that global optima are fairly tightly clustered when $\frac{N}{M} = 1$, so typical landscapes can still be roughly described as 'big valleys'. However, when $\frac{N}{M} = 1$ the degree of smoothing required to traverse these valleys is much higher than it is for the more extreme job:machine ratios.

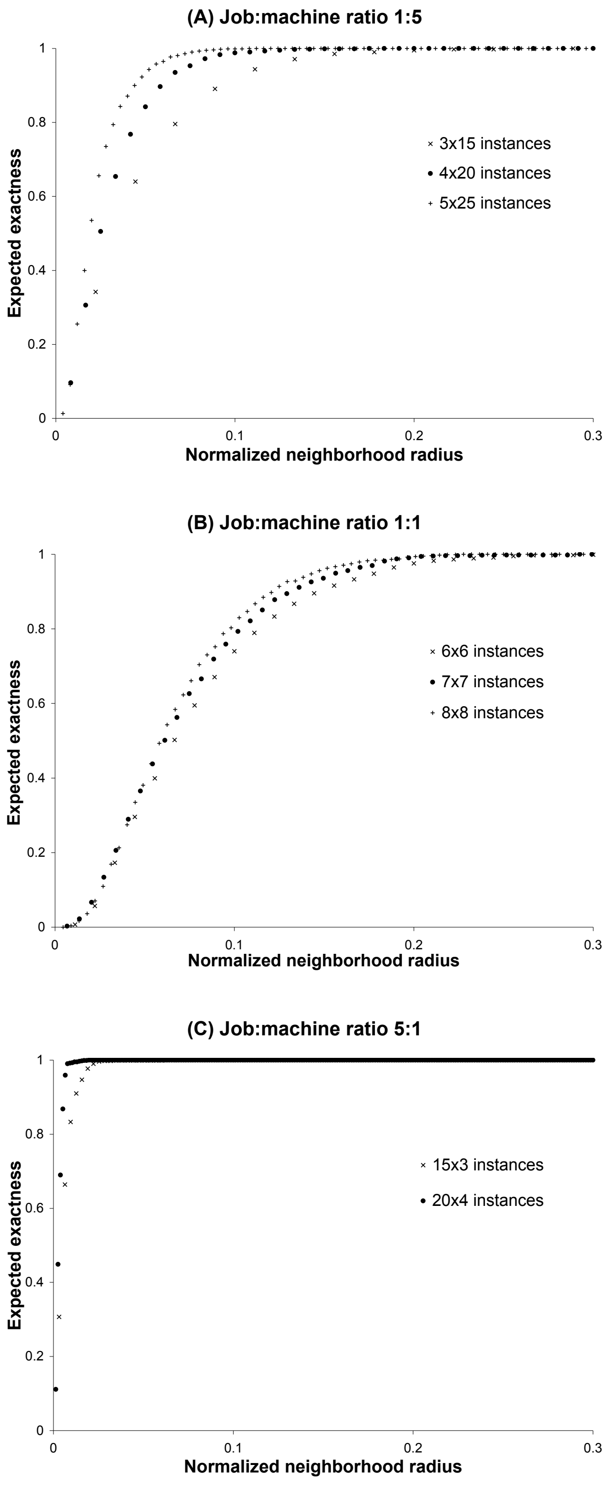

Figure 4: Expected exactness as a function of normalized neighborhood radius. Graphs (A), (B), and (C) depict curves for random instances with $\frac{N}{M} = \frac{1}{5}$, 1, and 5, respectively.

7. Limitations

There are two primary limitations of the experiments reported in this paper. First, we have collected statistics about small, random instances of the job shop scheduling problem. There is no guarantee that our observations will generalize to larger instances, or to structured problem instances (Watson *et al.* 2002). Second, we examined only *aggregate* behavior over ensembles of random instances, rather than behavior of particular instances. Aside from the restriction on instance size, none of these limitations are inherent in our approach.

8. Conclusions

Empirically, we have demonstrated that for low job:machine ratios, low-makespan schedules are clustered in a small region of the search space and many disjunctive edge orientations are common to all low-makespan schedules. As the job:machine ratio increases, low-makespan schedules become dispersed throughout the search space and there are no edge orientations common to all such schedules. For the two limiting values of the job:machine ratio (0 and ∞) we proved theorems that give insight into what happens in the general case. Finally, we examined neighborhood exactness in the JSSP, and discussed the implications of our results for the 'big valley' picture of JSSP search landscapes.

References

Balas, E. 1969. Machine sequencing via disjunctive graphs: An implicit enumeration algorithm. *Operations Research* 17:1–10.

Beasley, J. 1990. OR-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society* 41(11):1069–1072.

Boese, K. D.; Kahng, A. B.; and Muddu, S. 1994. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters* 16:101–113.

Brucker, P.; Jurisch, B.; and Sievers, B. 1994. A branch and bound algorithm for the job-shop scheduling problem. *Discrete Applied Mathematics* 49(1-3):107–127.

Fisher, H., and Thompson, G. L. 1963. Probabilistic learning combinations of local job-shop scheduling rules. In Muth, J. F., and Thompson, G. L., eds., *Industrial Scheduling*. Englewood Cliffs, NJ: Prentice-Hall. 225–251.

Glover, F., and Laguna, M. 1997. *Tabu Search*. Boston, MA: Kluwer Academic Publishers.

Jain, A., and Meeran, S. 1998. A state-of-the-art review of job-shop scheduling techniques. Technical report, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland.

Jones, A., and Rabelo, L. C. 1998. Survey of job shop scheduling techniques. Technical report, National Institute of Standards and Technology, Gaithersburg, MD.

Kim, Y.-H., and Moon, B.-R. 2004. Investigation of the fitness landscapes in graph bipartitioning: An empirical study. *Journal of Heuristics* 10:111–133.

Lourenço, H.; Martin, O.; and Stützle, T. 2003. Iterated local search. In Glover, F., and Kochenberger, G., eds., *Handbook of Metaheuristics*. Boston, MA: Kluwer Academic Publishers.

Martin, O. C.; Monasson, R.; and Zecchina, R. 2001. Statistical mechanics methods and phase transitions in combinatorial problems. *Theoretical Computer Science* 265(1-2):3–67.

Mattfeld, D. C.; Bierwirth, C.; and Kopfer, H. 1999. A search space analysis of the job shop scheduling problem. *Annals of Operations Research* 86:441–453.

Mézard, M., and Parisi, G. 1986. A replica analysis of the traveling salesman problem. *Journal de Physique* 47:1285–1296.

Monasson, R.; Zecchina, R.; Kirkpatrick, S.; Selman, B.; and Troyansky, L. 1999. Determining computational complexity from characteristic 'phase transitions'. *Nature* 400:133–137.

Nowicki, E., and Smutnicki, C. 1996. A fast taboo search algorithm for the job-shop problem. *Management Science* 42(6):797–813.

Nowicki, E., and Smutnicki, C. 2001. Some new ideas in TS for job shop scheduling. Technical Report 50/2001, University of Wroclaw.

Nowicki, E., and Smutnicki, C. 2002. Some new tools to solve the job shop problem. Technical Report 60/2002, University of Wroclaw.

Reeves, C. R., and Yamada, T. 1998. Genetic algorithms, path relinking, and the flowshop sequencing problem. *Evolutionary Computation* 6:45–60.

Roy, B., and Sussmann, B. 1964. Les problèmes dordonnancement avec contraintes disjonctives. Note D.S. no. 9 bis, SEMA, Paris, France, Décembre.

Slaney, J., and Walsh, T. 2001. Backbones in optimization and approximation. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001)*, 254–259.

Streeter, M. J., and Smith, S. F. 2005. Supplemental material for ICAPS 2005 paper 'Characterizing the distribution of low-makespan schedules in the job shop scheduling problem'. `http://www.cs.cmu.edu/~matts/icaps_2005`.

Taillard, E. 1994. Parallel taboo search techniques for the job shop scheduling problem. *ORSA Journal on Computing* 6:108–117.

van Laarhoven, P.; Aarts, E.; and Lenstra, J. 1992. Job shop scheduling by simulated annealing. *Operations Research* 40(1):113–125.

Watson, J.-P.; Beck, J. C.; Howe, A. E.; and Whitley, L. D. 2001. Toward an understanding of local search cost in job-shop scheduling. In Cesta, A., ed., *Proceedings of the Sixth European Conference on Planning*.

Watson, J.-P.; Barbulescu, L.; Whitley, L. D.; and Howe, A. 2002. Contrasting structured and random permutation flow-shop scheduling problems: search-space topology and algorithm performance. *INFORMS Journal on Computing* 14(2):98–123.

Appendix A: Proof of Lemma 2

We adopt the following notation. Where I is a JSSP instance, S is a schedule for I, and o is an operation in I,

- $est(o) \equiv$ the start time assigned to o by S
- $ect(o) \equiv est(o) + \tau(o)$
- $\mathcal{J}(o) \equiv$ the operation performed just before o in o's job
- $\mathcal{M}(o) \equiv$ the operation performed just before o on o's machine.

The choice of I and S will be apparent from context. We make use of the following well-known inequality.

Chernoff Bounds. *Let X be the sum of a set of independent, identically distributed (i.i.d) indicator variables (i.e., variables that always take on values in {0, 1}). Then for* $0 < \beta \leq 1$,

$$\mathbb{P}[|X - \mu| \geq \beta\mu] \leq e^{\frac{-\beta^2\mu}{3}}$$

Lemma 2. *Let M be constant and let $I_{N,M}$ be a random N by M JSSP instance. Then w.h.p (as $N \to \infty$) there exists an optimal schedule for $I_{N,M}$ with the property that no machine is idle until all operations performed on that machine have been completed.*

Proof. Let π be an random permutation of the jobs of $I_{N,M}$. Associate with each machine m a set of M queues, designated $q_{m,i}$ for $1 \leq i \leq M$. Let queue $q_{m,i}$ contain all operations performed on machine m as part of jobs whose i^{th} machine is m, with the operations listed in the order given by π. Let the schedule S specify that each machine m processes operations in the order given by the queues, starting with $q_{m,1}$ and proceeding up to $q_{m,M}$. We show that w.h.p S has the property described in the statement of the lemma.

We must first introduce two concepts. Let an operation o be *delayed* if $\mathcal{M}(o)$ exists and $est(o) \neq ect(\mathcal{M}(o))$. Let an operation o be an *earliest delayed operation* if o is delayed, and no operations with start time less than that of o are delayed. We now show that the probability that a randomly selected operation is an earliest delayed operation decreases exponentially with N.

Let o be a randomly selected operation, performed on machine m as part of a job J. Let m be the i^{th} machine used in J, and let αN jobs come before J in π. Assume that $\mathcal{M}(o)$ exists, and that $i > 1$ so $\mathcal{J}(o)$ also exists (otherwise o is delayed with probability 0). Further assume that no operations with start time earlier than that of o are delayed (otherwise o is an earliest delayed operation with probability 0). Under these assumptions, o is delayed iff. $ect(\mathcal{J}(o)) > ect(\mathcal{M}(o))$. Additionally, $ect(\mathcal{M}(o))$ is equal to the sum of the durations of

$$\left(\sum_{1\leq j\leq i-1} |q_{m,j}|\right) + |\{\bar{o} : \bar{o} \text{ precedes } o \text{ in } q_{m,i}\}| \qquad (1)$$

operations. Similarly, $ect(\mathcal{J}(o))$ is equal to the sum of

$$\left(\sum_{1\leq j\leq i-2} |q_{m',j}|\right) + |\{\bar{o} : \bar{o} \text{ precedes } \mathcal{J}(o) \text{ in } q_{m',i-1}\}| + 1 \qquad (2)$$

operation durations, where $\mathcal{J}(o)$ is performed on machine m'.

Let s and q denote the first and second terms of (1), respectively, while s' and q' denote the corresponding terms of (2) (we will ignore the $+1$ term in (2)). We want to bound the probability that $s+q-s'-q' < \frac{N}{2M}$. We do this by computing separate bounds on the probability that $s - s' \leq \frac{3N}{4M}$ and the probability that $q' - q \geq \frac{N}{4M}$.

First consider the probability that $q' - q \geq \frac{N}{4M}$. Both q' and q cannot exceed αN, so if $\alpha < \frac{1}{4M}$ this probability is 0. Assume $\alpha \geq \frac{1}{4M}$. Either q' or q can be viewed as the sum of αN i.i.d. zero-one indicator variables I_n, where $1 \leq n \leq \alpha N$. (In the case of q, I_n equals 1 iff. the n^{th} job in π uses m as its i^{th} machine.) $\mathbb{E}[q] = \mathbb{E}[q'] = \frac{\alpha N}{M}$, and we can apply Chernoff bounds with $\beta = \frac{1}{8}$ to obtain

$$\mathbb{P}[|q - \mathbb{E}[q]| \geq \frac{\alpha N}{8M}] \leq e^{\frac{-\alpha N}{192M}} \qquad (3)$$

and a similar inequality for q'. By Bonferroni's inequality and the fact that $\alpha \leq 1$, the probability that $q' - q \geq \frac{N}{4M}$ cannot exceed twice the right hand side of (3). By our assumption that $\alpha \geq \frac{1}{4M}$, this cannot exceed $2e^{\frac{-N}{768M^2}}$.

Similarly, $\mathbb{E}[s] = \frac{N(i-1)}{M}$ and $\mathbb{E}[s'] = \frac{N(i-2)}{M}$, so $\mathbb{E}[s\text{-}s'] = \frac{N}{M}$ for $i > 1$. Again we can use Chernoff bounds to bound the probability that $s - s' \leq \frac{3N}{4M}$. Putting these bounds together using Bonferroni's inequality, we find that the probability that $s + q - s' - q' < \frac{N}{2M}$ is at most U, where

$$U = 2\left(e^{\frac{-N}{192M^3}} + e^{\frac{-N}{768M^2}}\right). \qquad (4)$$

Assume fixed s, q, s' and q', with $s + q - s' - q' \geq \frac{N}{2M}$. Each operation belongs to just one queue, so its duration contributes to at most one of the variables in $\{s, q, s', q'\}$. Thus $ect(\mathcal{M}(o))$ and $ect(\mathcal{J}(o))$ are independent sums of the durations of $s+q$ and $s'+q'$ random operations, respectively. By the central limit theorem, $ect(\mathcal{M}(o))$ is Gaussian with mean $\mu(s + q)$ and variance $(s + q)\sigma^2$, while $ect(\mathcal{J}(o))$ is Gaussian with mean $\mu(s' + q')$ and variance $(s' + q')\sigma^2$. Thus $\mathbb{E}[ect(\mathcal{M}(o)) - ect(\mathcal{J}(o))]$ is at least $\frac{\mu N}{2M}$. Both $(s+q)$ and $(s' + q')$ are at most N, so the standard deviations of $ect(\mathcal{M}(o))$ and $ect(\mathcal{J}(o))$ are at most $\sigma\sqrt{N}$.

Thus, $ect(\mathcal{M}(o))$ will be $\geq ect(\mathcal{J}(o))$ so long as both $ect(\mathcal{M}(o))$ and $ect(\mathcal{J}(o))$ are no more than $\frac{\mu\sqrt{N}}{\sigma 4M}$ standard deviations below and above, respectively, their means. Each of these variables falls more than k standard deviations below (above) its mean with probability $\frac{1}{2} - \Phi(k)$, where Φ is the normal distribution function. The expression $\frac{1}{2} - \Phi(k)$ is $O(e^{\frac{-k^2}{2}})$. Putting these facts together, the probability that $ect(\mathcal{M}(o)) < ect(\mathcal{J}(o))$ is bounded by $U + \frac{1}{2} - \Phi(\frac{\mu\sqrt{N}}{\sigma 4M})$, which decreases exponentially with N.

Again using Bonferroni's inequality, the probability that *any* of the NM operations is an earliest delayed operation is at most $NM(U + \frac{1}{2} - \Phi(\frac{\mu\sqrt{N}}{\sigma 4M}))$. Thus S has the property described in the statement of the lemma w.h.p.

□