Constrainedness and the Phase Transition in Job Shop Scheduling

J. Christopher Beck

Dept. of Computer Science
University of Toronto
Toronto, Ontario,
CANADA, M5S 3G9
chris@cs.utoronto.ca

W. Ken Jackson

School of Computing Science
Simon Fraser University
Burnaby, B.C.,
CANADA, V5A 1S6
jackson@cs.sfu.ca

April 16, 1997

Abstract

Recent research has shown the existence of a "phase transition" for many combinatorial problems. In this paper we investigate the existence of a phase transition in the job shop scheduling problem. We apply standard estimates for the constrainedness of constraint satisfaction problems to job shop problems and describe empirical results identifying a phase transition at a constrainedness level of approximately 0.2. This phase transition point is much lower than expected, both from theory and in comparison with the phase transition point empirically found for other problems. We argue that this discrepancy is due to the weakness of the independence assumption used in the estimation of constrainedness.

1 Introduction

Recently, researchers have begun investigating the phase transition characteristics of a number of combinatorial problems including binary constraint satisfaction problems (CSPs), graph colouring, number partitioning, traveling salesman, and boolean satisfiability [Cheeseman et al., 1991; Gent and Walsh, 1994; Prosser, 1994 Smith and Dyer, 1994]. It has been observed that it is often easy to show that an over-constrained problem has no solution and also to find a solution to an under-constrained problem. Between these extremes, a problem may be critically constrained: it is difficult to either find a solution or show that none exists. The "phase transition" is an area in the problem space with a high density of critically constrained problems. The phase transition point can often be characterized with a particular problem parameter (*e.g.*, the ratio of clauses-to-variables for SAT [Mitchell et al., 1992]). Recently, Gent et al. have proposed a unified problem parameter, called constrainedness, and have shown how other parameters can be viewed as estimates of the constrainedness [Gent et al., 1996]. The constrainedness, κ, ranges from 0 to ∞ where 0 indicates an ensemble of problems that are completely under-constrained (*i.e.*, each problem has many solutions) and ∞ indicates an over-constrained ensemble of problems (*i.e.*, there are no solutions to any problem in the ensemble). A phase transition is expected to occur when $\kappa \approx 1$ and this has been empirically demonstrated on many problems.

In this paper we investigate the existence of a phase transition in job-shop scheduling. Our motivation is threefold:

1. **The development of an algorithm independent measure of difficulty of scheduling problems.** Such a development is a first step toward a deeper, empirical understanding of the behaviour of scheduling algorithms. If we can identify difficult problems and structural characteristics that correlate with this difficulty, we can begin to understand why specific heuristics perform well or fail to perform on various problems. With this knowledge, a further step is the design of algorithms that specifically address the difficult structural characteristics. While this is a long term research goal, an ability to first identify what are and are not difficult problems is critical.
2. **The application of the phase transition work to real world problems.** From a practical perspective, the ability to assess the difficulty of real world problems is a tremendous asset. Not only will we be better able to address such problems, we will also be able to better understand the similarities and differences between the real world problems and the research benchmarks based on them. We do not claim that job shop scheduling is a real world problem, though scheduling more generally clearly is. The investigation of a phase transition in the well understood and studied model of job shop is a first step towards more general classes of scheduling problems.
3. **The intuition that the constrainedness approach to the phase transition is not applicable to scheduling**. Previous work has associated the difficulty of solving job-shop scheduling instances with the presence of bottleneck resources [Sadeh, 1991] or on the interdependence between temporal and resource constraints [Fox, 1983]. As a single, extreme, data point for this intuition, consider that if a scheduling problem has either no resource constraints or no precedence constraints it is solvable in polynomial time [Garey and Johnson, 1979]. In contrast, estimates of constrainedness for CSPs are often made using an independence assumption: each constraint independently rules out some portion of the overall state-space. Our intuition, therefore, is that estimates of constrainedness for CSPs may not be applicable for job shop scheduling.

In this paper, we show that estimates of constrainedness developed for CSPs can be applied to job shop scheduling to reveal a phase transition. However, the phase transition point occurs at $\kappa \approx 0.20$, much lower than the phase transition point, $\kappa \approx 1$, predicted by theory or reported for other problems. We believe that this discrepancy occurs because job-shop problems violate the independence assumption to a greater extent than previously studied problems.

2 Constrainedness and the Phase Transition

[Gent et al., 1996] characterize the constrainedness of an ensemble of combinatorial problems and identify the phase transition with a factor, κ, defined by Equation (1).

$$\kappa \equiv 1 - \frac{\log(\langle Sol \rangle)}{N} \qquad (1)$$

Where $\langle Sol \rangle$ is the expected number of solutions averaged over the ensemble and N is the number of bits required to represent one state in the state-space[1]. A phase transition is expected to exist where $\kappa \approx 1$, that is, where the expected number of solutions is near zero.

1. All log() functions in this paper are assumed to be $\log_2()$.

For CSPs, [Gent et al., 1996] estimate the expected number of solutions for a single problem by a simple a characterization of the tightness of each constraint. Each constraint, *c*, on variables ($v_1, \ldots, v_a$), rules out a proportion, p_c, of the Cartesian product of the domains of the variables. To estimate the expected number of solutions, it is assumed that the states ruled out by each constraint are independent of those ruled out by any other constraint. On that basis $\langle Sol \rangle$ is estimated as follows:

$$\langle Sol \rangle = \left(\prod_{v \in V} m_v \right) \times \left(\prod_{c \in C} (1 - p_c) \right) \tag{2}$$

where:

- C is the set of constraints.
- V is the set of variables.
- m_v is the domain size of variable v.
- p_c is the tightness of constraint c as described above.

Equation (2) can be used to estimate $\langle Sol \rangle$ for each problem in the ensemble and the resulting values can be averaged to give an estimate of κ, using Equation (1).

Two points are worth emphasizing:

1. $\langle Sol \rangle$ is estimated using the independence assumption: the impact of each constraint on the number of solutions is independent of the impact of all other constraints.
2. κ is well-defined for a ensemble of problems, not for individual problems.

Though [Gent et al., 1996] are unclear about the definition of an ensemble, it appears that their operational definition is a set of problems generated by the same instances of the problem generation parameters. For random binary CSPs, [Gent et al., 1996] generate ensembles using the parameters *n*, *m*, p_1, and p_2, where *n* is the number of variables each with domain size *m*, p_1 is the constraint density (so that exactly $p_1 n(n\text{-}1)/2$ constraints are chosen) and p_2 is the tightness of each constraint (so that exactly $p_2 m^2$ pairs of values are eliminated from the Cartesian product of the domains). For a point in the parameter space, a set or ensemble of problems is generated. Since the parameters are constant for each problem instance in an ensemble, each problem has exactly the same state-space size, the same expected number of solutions, and therefore an estimate of κ can be made without explicit averaging over the problems in the ensemble.

As described later, we generate ensembles of problems such that each problem potentially has a different expected number of solutions and a different state-space size. Therefore, we estimate the expected number of solutions and the state-space size of each problem individually and use Equation (3) below to estimate κ for the ensemble.

$$\kappa \equiv 1 - \frac{\log\left(\dfrac{\sum_{i=1}^{k} \langle Sol_i \rangle}{k} \right)}{\log\left(\dfrac{\sum_{i=1}^{k} S_i}{k} \right)} \tag{3}$$

Where:

- $\langle Sol_i \rangle$ is the estimated number of solutions for problem *i* in the ensemble.

- S_i is the size of the state-space for problem *i*.
- *k* is the number of problems in the ensemble.

3 Job-Shop Scheduling

The traditional $n \times m$ job-shop scheduling problem is defined by *m* resources (machines) and *n* jobs. Each job consists of *m* totally ordered activities, one on each resource. Each activity:

- has a duration for which it must contiguously execute (*i.e.*, no preemption).
- requires a single resource for its entire duration.
- requires a different resource than each of the other activities in the same job.
- has precedence constraints that require that it must not start until all its predecessors have finished: the predecessors are defined by the total ordering on the activities in a job.

Each resource is limited to executing one activity at a time. Each job has a release-date, the time after which the activities in the job may be executed, and a due-date, the time by which the last activity in the job must finish.

A schedule is an assignment of a start time to each activity. We are asked to find the shortest possible schedule that satisfies all the precedence, resource, release-date, and due-date constraints. In this paper, we consider the decision version of the problem: the release-date of each job is 0, the due-date of each job is given by a common value *h*, and we are asked to find a schedule that satisfies all the constraints. Job-shop scheduling is NP-complete [Garey and Johnson, 1979].

A job-shop instance can be easily formulated as a binary CSP as follows:[2]

Constants:

- dur_A: the duration of activity *A*.
- *h*: the common due-date for all the jobs.

Variables:

- S_A: the start time of activity *A*.
- $dom(S_A)$: the domain of possible start-times for activity *A*. Initially, an activity can start at any time so $dom(S_A) = \{0, 1, \ldots, h\}$.
- est_A: the earliest possible start time of activity *A*. $est_A = \min(\mathrm{dom}(S_A))$
- lst_A: the latest possible start time of activity *A*. $lst_A = \max(\mathrm{dom}(S_A))$

Constraints:

- Precedence constraints (activity *A* must end before activity *B* starts): $S_A + dur_A \le S_B$ where activity *A* is ordered immediately before activity *B* in some job.
- Due-date constraints (the last activity in each job must end before the global due-date *h*): $S_A + dur_A \le h$ where activity *A* is the last activity in some job.
- Resource constraints: (activities *A* and *B* cannot execute at the same time): $(S_A + dur_A \le S_B) \vee (S_B + dur_B \le S_A)$ where activities *A* and *B* require the same resource.

3.1 Constraint Propagation as a Preprocessing Step

Initially, the domain of each start time variable is the set $\{0, 1, \ldots, h\}$. There are several constraint propagation schemes that can be applied to reduce the domains of these variables:

2. This is one formulation. An alternative formulation uses variables that indicate the ordering of two activities on the same resource.

- Temporal propagation enforces arc-B-consistency [Lhomme, 1993] on the precedence constraints. For example, enforcing arc-B-consistency on the constraint $S_A + 5 \le S_B$ with $dom(S_A) = \{1, 2, \ldots, 10\}$ and $dom(S_B) = \{1, 2, \ldots, 10\}$ will reduce the domain of S_A to the set {1, 2, 3, 4, 5} and the domain of S_B to the set {6, 7, 8, 9, 10}.
- Constraint-based analysis [Erschler et al., 1976; Erschler et al., 1980] helps to resolve the disjunction in resource constraints. If one of the disjuncts in a resource constraint can be shown to be false then the other disjunct must be true and it can be asserted as a new precedence constraint between activities on the same resource.
- Edge-finding [Carlier and Pinson, 1989; Nuijten et al., 1993] is like constraint-based analysis in that it helps to resolve the disjunction in resource constraints. Edge-finding adds unary temporal constraints, enforcing new upper and lower bounds on the start and end times of activities. These unary constraints are inferred from an examination of the subsets of activities on a single resource while constraint-based analysis just considers pairs of activities.

We have experimented with combinations of these methods resulting in the following sequence of increasingly powerful propagation schemes:

1. No propagation.
2. Temporal propagation.
3. Temporal propagation plus constraint-based analysis.
4. Temporal propagation plus constraint-based analysis plus edge-finding.

Applying constraint propagation as a preprocessing step reduces the domains of variables and therefore affects the estimates of κ. Given the intuition that the difficulty of job shop scheduling is largely due to interactions between the constraints, our hope is that applying a powerful constraint propagation scheme might account for some of the interactions and thus improve the estimate of κ.

4 Constrainedness and the Job Shop

Our approach is to estimate the constrainedness for an ensemble of job shop problems by formulating each problem as a binary CSP, using Equation (2) to estimate the number of solutions, and using Equation (3) to estimate κ. In this section, we describe the details of this approach related to the following. First, Equation (2) requires knowing the tightness p_c of each constraint. We describe how we determine the constraint tightness for each type of constraint in a job shop problem. Second, the tightness of each constraint depends on how much constraint propagation is done.

Intuitively, one method to calculate the tightness of a constraint is to generate the compatibility matrix for the constraint and then count the number of zeroes. Figure 1 shows some examples of compatibility matrices for the constraints in a job shop problem.

In Equations (4) through (6) we present simple equations that calculate the proportion of zeros in each constraint matrix based on the form of the constraint.

The tightness of a precedence constraint $S_A + dur_A \le S_B$ was calculated as follows:

$$p_{(S_A + dur_A \le S_B)} = 1 - \frac{((h+1) - dur_A)((h+1) - dur_A + 1)}{2(h+1)^2} \qquad (4)$$

$$
S_A \begin{array}{c|ccccccc}
 & \multicolumn{7}{c}{S_B} \\
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\qquad
S_A \begin{array}{c|ccccccc}
 & \multicolumn{7}{c}{S_B} \\
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
6 & 1 & 1 & 1 & 0 & 0 & 0 & 0
\end{array}
\qquad
S_A \begin{array}{c|c}
 & h \\
 & 0 \\
\hline
0 & 1 \\
1 & 1 \\
2 & 1 \\
3 & 1 \\
4 & 0 \\
5 & 0 \\
6 & 0
\end{array}
$$

Precedence Constraint $S_A + 3 \leq S_B$ — Resource Constraint $(S_A + 3 \leq S_B) \vee (S_B + 4 \leq S_A)$ — Due-Date Constraint $S_A + 3 \leq S_B$

Figure 1. Example Compatibility Matrices for Job Shop Constraints

Note that we exclude the implicit precedence constraints that arise due to the transitivity of the precedence relation. The transitive constraints clearly violate the independence assumption: they can not be violated unless another immediate precedence constraint is also violated.

The due-date constraints are similar to precedence constraints but have a single value rather than a variable on the right-hand-side of the inequality. The tightness of a due-date constraint $S_A + dur_A \leq h$ was calculated from the following equation:

$$p_{(S_A + dur_A \leq h)} = dur_A / (h + 1) \tag{5}$$

Equation (6) below was used to calculate the tightness of a resource constraint $(S_A + dur_A \leq S_B) \vee (S_B + dur_B \leq S_A)$:

$$p_{((S_A + dur_A \leq S_B) \vee (S_B + dur_B \leq S_A))} = p_{(S_A + dur_A \leq S_B)} + p_{(S_B + dur_B \leq S_A)} - 1 \tag{6}$$

4.1 Estimating Constrainedness with Propagation

Constraint propagation revises the domains of the start time variables and so some of the simplifying assumptions (*e.g.*, the domains of all variables are $\{0, \ldots, h\}$) that were used to derive Equations (4) through (6) do not hold when constraint propagation is done. We used the following equations when constraint propagation was applied.

The tightness of a precedence constraint was calculated as follows:

$$p_{(S_A + dur_A \leq S_B)} = \frac{\sum_{t = est_B}^{lst_B} max(0, lft_A - t)}{|dom(S_A)| \times |dom(S_B)|} \tag{7}$$

Because we have performed temporal propagation and there is a precedence constraint between activities *A* and *B*, we are guaranteed that $est_A \leq est_B$, which is necessary to ensure that we do not over-count the possible start times of activity *A*.

The tightness of the due-date constraints is zero because propagation removes any start-times that do not conform to the activity ending at or before the due-date *h*. Therefore, the tightness of a due-date constraint has no effect on the estimate of the expected number of solutions.

The tightness of a resource constraint was modified to be the following:

$$p_{((S_A + dur_A \le S_B) \vee (S_B + dur_B \le S_A))} = \frac{\sum_{t = est_B}^{t = lst_B} max(0, min(lst_A, t + dur_B - 1) - max(est_A, t - dur_A + 1) + 1)}{|dom(S_A)| \times |dom(S_B)|} \quad (8)$$

When using constraint-based analysis, new precedence constraints are deduced in place of some of the disjunctive resource constraints. In these cases, we included just the derived precedence constraint and excluded the corresponding resource constraint from the estimation of the expected number of solutions.

5 Experimental Results

We ran a series of computational experiments to empirically determine the existence and location of a phase transition in job shop scheduling problems. This section describes these experiments including how we generated problem ensembles, empirical results on the location of the phase transition point with respect to estimates of κ (with and without constraint propagation as a preprocessing step), and the location of the phase transition point with respect to the actual value of κ on small problems.

5.1 Generation of Ensembles of Job Shop Instances

We generated 120 problem ensembles each containing 100 instances of the job shop scheduling problem, as described below. Each instance in an ensemble has the same number of jobs, the same number of resources, and the same “horizon factor”. The horizon factor f specifies by what factor the due-date for each job exceeds a lower bound on the schedule’s duration. That is, for a horizon factor f and lower bound b on the schedule’s duration, the due-date $f \times b$ was used. We calculated lower bounds following [Taillard, 1993].

We generated random job shop instances using Taillard’s job shop generator [Taillard, 1993]. In this generator, the following parameters, as well as a pair of random seeds is used to generate each instance.

- n: the number of jobs was varied between 3 to 12 as described below
- m: the number of resources was varied between 3 to 15 as described below.

We generated two sets of problems based on size. In the first set all problems were square, that is,

$$n = m, \text{ and } n \in \{3, 5, 6, 8, 10, 12\}$$

The problems in the second set were not square and consisted of instances with the following combinations of n and m:

$$5 \times 10,\ 5 \times 15,\ 8 \times 12,\ 10 \times 5,\ 10 \times 15,\ 12 \times 8$$

For each problem size, we generated 100 problems using a different pair of random seeds (as required by Taillard generator). The due-dates for these problems were set using the following horizon factors:

0.9, 0.967, 1.033, 1.1, 1.167, 1.233, 1.3, 1.367, 1.433, 1.5

Note that the first two horizon factors (0.9 and 0.967) necessarily produce ensembles such that all the problems are over-constrained.

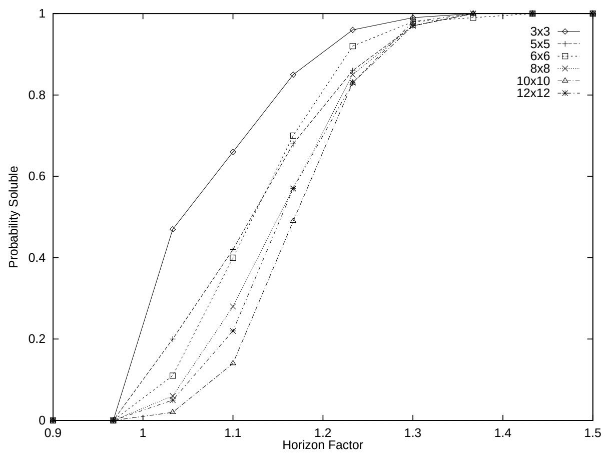

Figure 2. Solubility vs. Horizon Factor

5.2 The Phase Transition for Estimates of κ

We solved each problem using the NumODO scheduler [Beck et al., 1997b; Beck et al., 1997a], that is, for each instance, NumODO was able to either find a solution or was able to show that the instance was not satisfiable (by exhaustive search).3

The horizon factor is naturally related to constrainedness (*i.e.*, the smaller the horizon factor the more constrained the problem). In Figure 2, we plot the probability of solubility against the horizon factor to examine the possibility of a phase transition characterized simply by the horizon factor. The point where 50% the problems are satisfiable occurs when the horizon factor is between 1.0 and 1.3 but there is no strong indication of a phase transition with respect to the horizon factor.

The graphs in Figure 3 and Figure 4 show the probability of solution versus the estimated constrainedness for the square problems. Figure 3 estimates the constrainedness without any constraint propagation while Figure 4 uses temporal constraint propagation. The graphs exhibit the classical phase transition behaviour, however, the phase transition point is quite low, at $\kappa \approx 0.20$ when no propagation is done. Estimating constrainedness using temporal propagation shifted the phase transition point to $\kappa \approx 0.30$.

Adding more powerful constraint propagation schemes, constraint-based analysis and constraint-based analysis plus edge-finding, resulted in little change in the phase transition point from simply using temporal propagation. Graphs of solubility versus constrainedness for these methods are presented in Figure 5 and Figure 6, respectively. Note that the horizontal line in the graph in Figure 6 occurs because of an interpolation anomaly. The edge-finding propagation technique is able to show that all the problems in the ensembles created with the smallest two horizon factors are unsatisfiable. Since the constrainedness is estimated after running edge-finding, the κ value for these ensembles is infinity and the probability of having a solution is zero. The horizontal line is a linear interpolation between the previous data point and the point $(\infty, 0)$.

3. The exponential search required to prove that a problem has no solutions prevented the use of larger problems in our experiments.

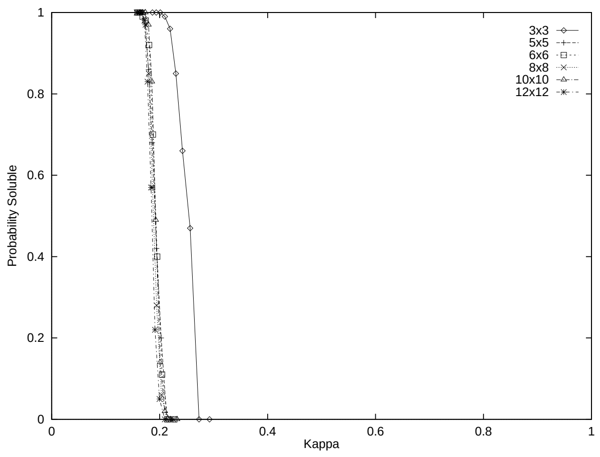

Figure 3. Phase Transition with No Propagation

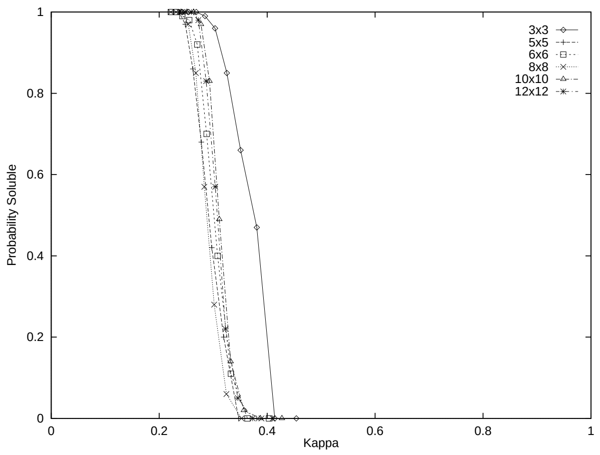

Figure 4. Phase Transition with Temporal Propagation

The results for the non-square problems were similar and are shown in Figure 7 for the case where temporal propagation was done. The phase transition is evident but it is not as sharp as with the square problems.

5.3 The Phase Transition for Actual κ

Our intuition is that the phase transition point is low because the *estimates* of κ did not adequately take in to account the dependencies between constraints even with constraint propagation. An alternative explanation is that the theory behind constrainedness does not apply to job shop scheduling. To investigate this, we calculated the actual constrainedness for the 3×3 ensembles by modifying the NumODO scheduler to find all solutions. The actual number of solutions was then used to

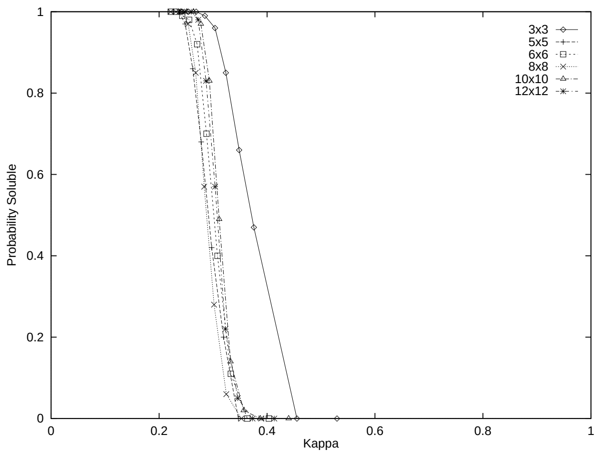

Figure 5. Phase Transition with Temporal Propagation + CBA

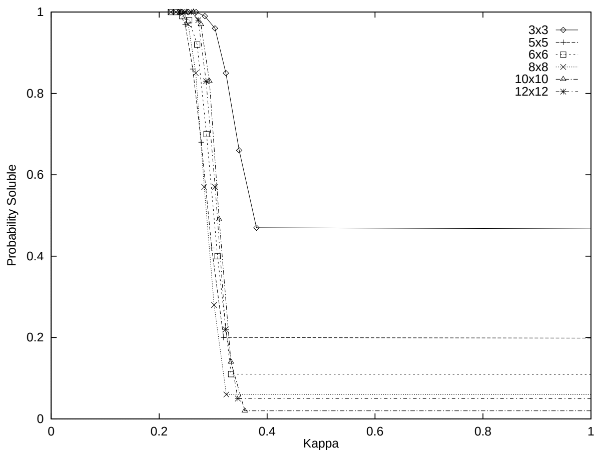

Figure 6. Phase Transition with Temporal Propagation + CBA + Edge-Finding

compute the actual value for κ according to Equation (3). The results, shown in Figure 8, reveal a phase transition point (where 50% of the problems are satisfiable) when the actual κ is approximately 1.0.

Again, the horizontal line occurs because of the interpolation anomaly with a data point at $(\infty, 0)$. Unfortunately, the computational time required to find all solutions prevented us from computing the actual κ values for ensembles of larger problems.

6 Discussion

We have observed a phase transition in job shop scheduling at $\kappa \approx 0.20$, far less than both the theoretical phase transition point and that empirically demonstrated for other problems [Gent et al., 1996]. We believe that this discrepancy occurs because

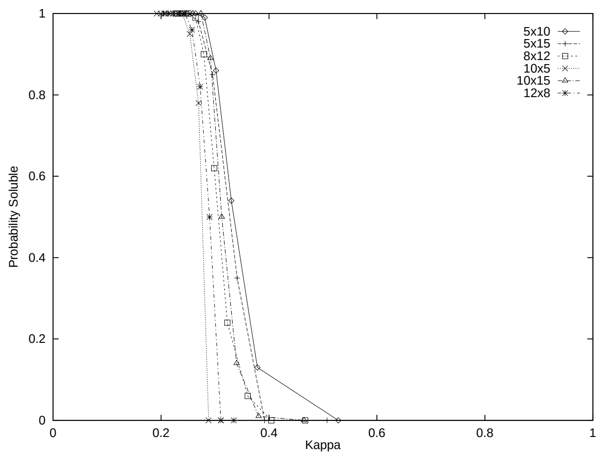

Figure 7. Non-Square Problems with Temporal Propagation

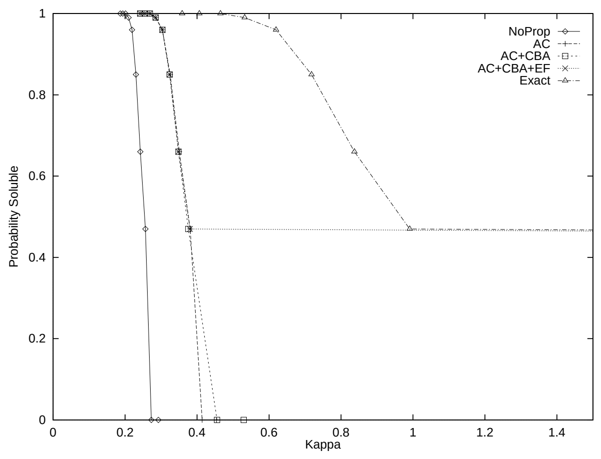

Figure 8. Phase Transition vs. Actual κ for 3x3 Problems

the independence assumption made to estimate κ does not hold for job shop scheduling: the states ruled out by a constraint are not independent of those ruled out by other constraints.

Our experimental results suggest that, for job shop scheduling, estimating κ using the independence assumption underestimates the actual value of κ. In other words, there are some interactions among the constraints that rule out more solutions than the individual constraints acting alone. In particular, we have shown that for small problems, where all solutions could be enumerated and, therefore, where κ could be precisely calculated, the phase transition occurs at $\kappa \approx 1.0$, as predicted and expected. These results support the conclusion that it is the estimate of κ that is at fault rather than some property of the job shop scheduling problem that invalidates the theory of constrainedness.

While our experimental results are consistent with our intuition vis-a-vis the independence assumption, they do not confirm it. We have not yet shown that it is dependencies among constraints that cause our estimates of κ to be low.

The results of this experiment raise a number of interesting points with respect to constrainedness in general and the constrainedness of job shop scheduling as compared to that of other problem classes.

- In their paper, [Gent et al., 1996] note that, empirically, different classes of problems exhibit a phase transition at varying values of κ. For example, the phase transition in 3-SAT occurs at $\kappa \approx 0.82$, in 3-coloring at $\kappa \approx 0.84$, and in number partitioning at $\kappa \approx 0.96$. [Gent et al., 1996] suggest that this indicates "number partitioning problems at the phase transition may in some sense be more constrained." While it is not clear precisely what this statement means, it would seem to imply that job shop scheduling, given our results of a phase transition point of $\kappa \approx 0.20$, is much *less* constrained than these other problem classes. An alternative interpretation is that independence assumption used in the estimation of constrainedness holds to different degrees in different problem classes. In problem classes with significant interaction amongst the constraints, the phase transition point would be observed at a lower value of κ.
- The low value of κ at the phase transition point indicates that proportionately more solutions in job shop scheduling are ruled out by constraint interaction than, for example, in graph colouring. This supports the notion that, as compared to graph colouring, more of the difficulty in job shop scheduling comes from the interactions of the constraints. This different source of difficulty suggests that these problem classes have important differences that are obscured when they are simply viewed from the perspective of their NP-hardness. In particular, heuristic solution strategies for graph colouring may not be appropriate for adaptation to job shop scheduling, and vice versa.
- In general, if the independence assumption does not hold for some problem class, the estimate of κ may be either an overestimate or an underestimate. That is, we may be able to construct examples where constraints together rule out more solutions than the sum of the individual contributions and conversely we can construct examples where constraints overlap in the solutions that they rule out. This leads to a prediction that for problems with redundant or overlapping constraints, the phase transition should be observed at a $\kappa > 1$. To our knowledge, this has not been observed for any problem class.

The prime area for further research is to test our intuition regarding the independence assumption: can we empirically evaluate the independence (or lack thereof) of constraints in a job shop problem? Are there dependencies between resource and precedence constraints? A more accurate estimate for κ must explicitly take into account some of the interactions between the constraints. Two possibilities are the following:

1. [Hooker and Vinay, 1995] provide a second-order estimate of the probability that a SAT problem has a solution. This estimate takes into account interactions between pairs of clauses in a SAT problem. It may be fruitful to apply their methods to job shop problems.
2. As noted in our introduction, it is believed in the scheduling community that problem structures such as bottleneck resources (*i.e.*, a resource that needs to execute a large number of activities in a short period of time) contribute significantly to problem difficulty [Sadeh, 1991]. Indeed, we can extend the notion of bottleneck to a set of resources such that a set of tightly temporally related activities

must execute in a short period of time on a small set of resources[4]. Given a graph-based representation of a scheduling problem, it may be possible to use some graph measurements (*e.g.*, minimum cut-set) to improve the estimation of κ.

7 Conclusion

This work is the first demonstration of a phase transition for the job shop scheduling problem. We empirically found a phase transition at $\kappa \approx 0.20$ when no constraint propagation was done and at $\kappa \approx 0.30$ when constraint propagation was done as a preprocessing step. The phase transition at $\kappa \approx 0.30$ was found to be relatively independent of whether simple temporal propagation or more powerful propagation (constraint-based analysis and edge-finding) was done.

The disparity between the observed κ value at the phase transition for job shop scheduling and both the theoretically predicted $\kappa \approx 1$ and the empirically observed κ for other problems [Gent et al., 1996] can be accounted for by postulating significant interaction among the constraints. This postulate is supported by the intuition in the scheduling community that much of the difficulty in scheduling arises from interactions between constraints.

This work highlights the intuition that the independence assumption may hold to different degrees in different problem classes. Those classes, such as job shop scheduling, where there is a tight interaction between constraints will exhibit a phase transition at a lower value of κ while those with independent or redundant constraints will exhibit a phase transitions near or above $\kappa \approx 1$. These intuitions explain our results but remain to be more fully investigated.

Acknowledgements

This research was funded in part by the Natural Science and Engineering Research Council of Canada, Numetrix Limited, the Institute for Robotics and Intelligent Systems, the Manufacturing Research Corporation of Ontario, and Digital Equipment of Canada. This work originated at the “Intelligent Scheduling Mini-Workshop” and we wish to thank the following participants for the interesting discussions and insights: Andrew Davenport, Daya Gaur, Morten Irgens, and Qi Zhao. We would also like to thank Mark Fox and Bill Havens for their comments on and discussions of earlier drafts of this paper.

8 References

Beck, J. C., Davenport, A. J., Sitarski, E. M., and Fox, M. S. (1997a). Beyond contention: extending texture-based scheduling heuristics. In *Proceedings of AAAI-97*. AAAI Press, Menlo Park, California.

Beck, J. C., Davenport, A. J., Sitarski, E., and Fox, M. S. (1997b). Texture-based heuristics for scheduling revisited. In *Proceedings of AAAI-97*. AAAI Press, Menlo Park, California.

Carlier, J. and Pinson, E. (1989). An algorithm for solving the job-shop problem. *Management Science*, 35(2):164–176.

Cheeseman, P., Kanefsky, B., and Taylor, W. (1991). Where the really hard problems are. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, volume 1, pages 331–337.

4. Anecdotally, some hard problems (such as the notorious FT10 [Carlier and Pinson, 1989]) demonstrate such problem structure.

Erschler, J. , Roubellat, F. , and Vernhes, J. P. (1976).Finding some essential characteristics of the feasible solutions for a scheduling problem. *Operations Research*, 24:772–782.

Erschler, J. , Roubellat, F. , and Vernhes, J. P. (1980). Characterising the set of feasible sequences for n jobs to be carried out on a single machine. *European Journal of Operational Research*, 4:189–194.

Fox, M.S., (1983). Constraint-Directed Search: A Case Study of Job-Shop Scheduling. Ph .D. Thesis, Carnegie Mellon University, CMU-RI-TR-85-7, Intelligent Systems Laboratory,ackn The Robotics Institute.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.

Gent, I. P., MacIntyre, E., Prosser, P., and Walsh, T. (1996). The constrainedness of search. In *Proceedings of AAAI-96*, volume 1, pages 246–252.

Gent, I. P. and Walsh, T. (1994). Computational phase transitions in real problems. Technical Report 724, Department of Artificial Intelligence, University of Edinburgh.

Hooker, J. N. and Vinay, V. (1995). Branching rules for satisfiability. *Journal of Automated Reasoning*, 15:359–383.

Lhomme, O. (1993). Consistency techniques for numeric CSPs. In *Proceedings of IJCAI-93*, volume 1, pages 232–238.

Mitchell, D., Selman, B., and Levesque, H. (1992). Hard and easy distributions of SAT problems. In *Proc, 10th National Conference on Artificial Intelligence*, pages 459–465.

Nuijten, W. P. M., Aarts, E. H. L., van Arp Taalman Kip, D. A. A., and van Hee, K. M. (1993). Randomized constraint satisfaction for job shop scheduling. In *Proceedings of the IJCAI'93 Workshop on Knowledge-Based Production, Scheduling and Control*, pages 251–262.

Prosser, P. (1994). Binary constraint satisfaction problems: Some are harder than others. In *11th European Conference on Artificial Intelligence*, pages 95–99.

Sadeh, N. (1991). *Lookahead techniques for micro-opportunistic job-shop scheduling*. PhD thesis, Carnegie-Mellon University. CMU-CS-91-102.

Smith, B. M. and Dyer, M. E. (1994). Locating the phase transition in constraint satisfaction problems. Technical Report 94.16, Division of Artificial Intelligence, University of Leeds. To appear in AI Journal.

Talliard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64:278–285.