AAAI - 93

Slack-Based Heuristics For Constraint Satisfaction Scheduling *

Stephen F. Smith
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sfs@isl1.ri.cmu.edu

Cheng-Chung Cheng
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
ccen@isl1.ri.cmu.edu

Abstract

In this paper, we define and empirically evaluate new heuristics for solving the job shop scheduling problem with non-relaxable time windows. The hypothesis underlying our approach is that by approaching the problem as one of establishing sequencing constraints between pairs of operations requiring the same resource (as opposed to a problem of assigning start times to each operation) and by exploiting previously developed analysis techniques for limiting search through the space of possible sequencing decisions, simple, localized look-ahead techniques can yield problem solving performance comparable to currently dominating techniques that rely on more sophisticated analysis of resource contention. We define a series of attention focusing heuristics based on simple analysis of the temporal flexibility associated with different sequencing decisions, and a similarly motivated heuristic for determining how to sequence a given operation pair. Performance results are reported on a suite of benchmark problems previously investigated by two advanced approaches, and our simplified look-ahead analysis techniques are shown to provide comparable problem solving leverage at reduced computational cost.

Introduction

In this paper, we propose and evaluate the performance of new look-ahead heuristics for solving the job shop scheduling problem with non-relaxable time windows. The problem originates from the manufacturing domain, and, as classically defined, involves synchronization of the production of N jobs in a facility with M machines. The production of a given job requires the execution of a sequence of operations (its process plan in manufacturing parlance). Each operation has a specified processing time and its execution requires the exclusive use of a designated machine for the duration of its processing (i.e. machines have unit processing capacity). Each job has an associated ready time and a deadline, and its production must be accomplished within this interval. The problem can be extended in various ways - to include selection among designated resource alternatives for each operation, to associate multiple resource requirements (e.g. machine, operator) with operations, etc. In any case, the objective is to determine a schedule for production that satisfies all temporal and resource capacity constraints.

The job shop scheduling with non-relaxable time windows problem is known to be NP-Complete (Garey & Johnson 1979). Accordingly, the development of effective heuristic procedures for solving this constraint satisfaction problem (CSP) has been the subject of considerable previous research. This work, with few exceptions, has sought to exploit the special structure of the problem, in particular the structure of resource capacity constraints, to enhance consistency enforcement and early search space pruning capabilities, to support more-informed backtracking, and to focus attention in elaborating the search (our principal interest in this paper). Most frequently, the job shop scheduling problem has been formulated as one of finding a consistent assignment of start times for each operation of each job (Johnston 1990), (Keng & Yun 1989), (Minton et al. 1990), (Sadeh 1991), and (Zweben et al. 1990). Here, the development of focus of attention (or variable ordering) heuristics has focused fairly exclusively on use of contention-based metrics. One recent approach which has produced strong comparative experimental results, relies on a dynamic variable ordering heuristic that maintains profiles of resource demand over time, repeatedly identifies the resource and time period of greatest expected contention, and focuses attention on scheduling the operation that contributes most to this "bottleneck" (Sadeh 1991).

A smaller number of efforts have alternatively treated the problem as one of posting sufficient additional sequencing constraints between pairs of operations contending for the same resource so as to ensure feasibility with respect to time and capacity con-

*This research reported in this paper has been sponsored in part by DARPA under contract F30602-90-C-0119, NASA under contract NCC 2-707, and the Robotics Institute.

straints. The solutions generated in this way typically represent a set of feasible schedules (i.e., the sets of operation start times that remain consistent with posted sequencing constraints), as opposed to a single assignment of operation start times. In (Erschler et al. 1976, 1980) the structure of resource capacity constraints is exploited to define dominance conditions for pruning the set of feasible sequencing alternatives at each stage of the search. More recently, (Muscettola 1993) has demonstrated the utility of global resource capacity analysis techniques (similar in spirit to the approach in (Sadeh 1991)) as a focusing mechanism within this alternative search space; in this case sequencing constraints are repeatedly posted between sets of conflicting operations until resource capacity analysis indicates no further possibility of resource contention.

Like (Muscettola 1993), we believe that the inherent flexibility gained by providing sets of feasible solutions offers considerable pragmatic value over typically overconstrained fixed times solutions. The principal claim of this paper, however, is that this second formulation of the problem also provides a more convenient search space in which to operate. When the problem is cast as a search for orderings between pairs of operations vying for the same resource, we argue that it is possible to obtain the look-ahead benefits of global resource capacity analysis through the use of simpler, local analysis of the sequencing possibilities associated with unordered operation pairs. We define a series of variable ordering heuristics based on measures of temporal slack which, when integrated with the search space pruning techniques developed in (Erschler et al. 1976), are shown to yield comparable problem solving performance to contention-based heuristics at a fraction of the computational cost.

The remainder of the paper is organized as follows. In Section 2, we specify the problem as a CSP search for operation pair orderings, and review dominance conditions that enable search space pruning relative to this model. In Sections 3 through 5, we propose a series of variable ordering heuristics and present comparative results on a previously studied suite of 60 test problems. Finally, in Section 6, we outline current work in applying the approach to schedule optimization.

Problem Representation and Search Framework

In more precise terms, a solution to the basic job shop scheduling CSP requires a consistent assignment of values to start time variables st_i for each operation i, under the following constraints:

- **sequencing restrictions** - for every precedence relation $i \rightarrow j$ specified between operations i and j in the process plan of a given job $\mathcal{J}$, $st_i + p_i \leq st_j$, where p_i is the processing time required by operation i of job $\mathcal{J}$.
- **resource capacity constraints** - for any two operations i and j requiring the same resource, $st_i + p_i \leq st_j \vee st_j + p_j \leq st_i$
- **ready times and deadlines** - for each operation i of job $\mathcal{J}$, $r_{\mathcal{J}} \leq st_i$ and $st_i + p_i \leq d_{\mathcal{J}}$, where $r_{\mathcal{J}}$ and $d_{\mathcal{J}}$ are the ready time and deadline respectively associated with job $\mathcal{J}$.

While this problem representation provides a direct basis for problem solving search (and in fact has been taken as the starting point of most previous research), the problem can be alternatively formulated as one of establishing sequencing constraints between pairs of operations contending for the same resource over time. In this case, we define a decision variable $ordering_{i,j}$ for each pair of operations i and j that require the same resource, which can take on either of two values: $i \rightarrow j$ (implying the constraint $st_i + p_i \leq st_j$) and $j \rightarrow i$ (implying $st_j + p_j \leq st_i$). A solution then is a consistent assignment of values to all ordering variables. There are several potential advantages to this formulation. The advantage emphasized in this paper is that the simpler structure of the search space enables more straightforward accounting of resource capacity constraints and the use of simpler, localized analysis of current solution structure as a basis for variable and value ordering.

Our problem solving framework assumes a backtrack search procedure in which the solution is incrementally extended through the repeated selection and binding of an as yet unconstrained $ordering_{i,j}$ variable (referred to as the posting of a new precedence relation). Whenever a new precedence relation is posted, constraint propagation is performed to ensure continued temporal consistency and maintain current bounds on the earliest start time and latest finish time of each operation. [1] If the decision $i \rightarrow j$ is taken, for example, then est_j (the earliest start time of j) and lft_i (the latest finish time of i) are updated by

$$est_j = \max\{est_j, est_i + p_i\}, \text{ and} \tag{1}$$

$$lft_i = \min\{lft_i, lft_j - p_j\}, \tag{2}$$

and these new values are then propagated forward or backward respectively through all pre-specified and posted temporal precedence relations. If during this process, $est_k + p_k$ becomes greater than lft_k for any operation k then an inconsistent set of assignments has been detected.

As indicated at the outset, our approach to directing the search integrates a procedure previously developed by Erschler *et al*, referred to as *Constraint-based Analysis (CBA)*, which exploits dominance conditions to prune the space of possible ordering assignments. To summarize their basic idea, assume that est_i and lft_i

[1] Since we are assuming in this paper that operation processing times are fixed, we could equivalently reason in terms of earliest and latest start times.

designate the current earliest start time and latest finish time respectively of a given operation i. Then, for any unordered pair of operations, i and j, contending for a particular resource, we can distinguish four different cases:

1. If $lft_i - est_j < p_i + p_j \leq lft_j - est_i$ then i must be scheduled before j in any feasible extension of the current ordering decisions. (case 1)
2. If $lft_j - est_i < p_i + p_j \leq lft_i - est_j$ then j must be scheduled before i in any feasible extension of the current ordering decisions. (case 2)
3. If $p_i + p_j > lft_j - est_i$ and $p_i + p_j > lft_i - est_j$ then there is no feasible schedule. (case 3)
4. If $p_i + p_j \leq lft_j - est_i$ and $p_i + p_j \leq lft_i - est_j$ then either sequencing decision is still possible. (case 4)

These dominance conditions of course provide only necessary conditions for determining a set of feasible schedules, and thus interleaved application of CBA and temporal constraint propagation yields an underspecified search procedure. What is needed to generate solutions are heuristics for resolving the undecided states specified in case 4. In this regard, previous use of CBA has emphasized fuzzy integration of sets of different scheduling rules. In (Bensana & Dubois 1988), a voting procedure based on fuzzy set theory and approximate reasoning was developed and used in conjunction with a set of fuzzy scheduling rules. In (Kerr & Walker 1989), fuzzy arithmetic together with fuzzy scheduling rules was utilized instead. Our goal, alternatively, is to investigate the effectiveness of CBA in conjunction with simple look-ahead analysis of current ordering flexibility. This leads to the search procedure that is graphically depicted in Figure 1, which we will refer to as precedence constraint posting (PCP). In the following sections, we define and evaluate a specific set of variable and value ordering heuristics.

Exploiting Estimates of Sequencing Flexibility

Intuitively, in situations where CBA leaves the search in a state with several unresolved ordering assignments (i.e., for each unordered operation pair, both ordering decisions are still feasible), we would like to focus attention on the ordering decision that is currently most constrained. Since the posting of any sequence constraint is likely to further constrain other ordering decisions that remain to be made, delaying the currently most constrained decision increases the chances of arriving at an infeasible problem solving state.

Implementation of such a variable ordering strategy requires a means of estimating the current flexibility associated with a given unresolved ordering decision. One simple indicator of flexibility is the amount of temporal slack that is retained by a given operation pair if a decision to sequence them is taken. To this end,

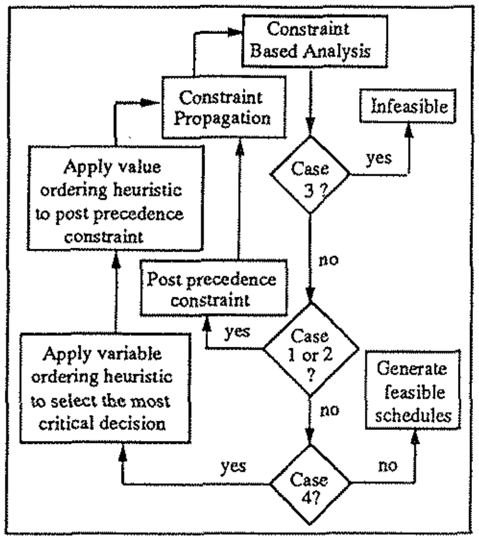
Constraint Based Analysis
Constraint Propagation
Infeasible
Apply value ordering heuristic to post precedence constraint
Case 3 ?
yes
no
Post precedence constraint
yes
Case 1 or 2 ?
Apply variable ordering heuristic to select the most critical decision
Generate feasible schedules
no
yes
Case 4?
no

Figure 1: PCP Search Procedure

we define two measures, corresponding to the two possible decisions that might be taken. For a given pair of currently unordered operations (i,j) contending for the same resource, we define the "temporal slack remaining after sequencing i before j" as

$$slack(i \rightarrow j) = lft_j - est_i - (p_i + p_j), \qquad (3)$$

and similarly the "temporal slack remaining after sequencing j before i" as

$$slack(j \rightarrow i) = lft_i - est_j - (p_i + p_j). \qquad (4)$$

Figure 2 provides a graphic illustration of $slack(i \rightarrow j)$ and $slack(j \rightarrow i)$. Note that in either case the remaining slack is shared by both i and j. Thus, the larger the temporal slack, the greater the chance that subsequent ordering decisions involving i and j can be feasibly imposed.

Given these measures of temporal slack, we now have a basis for identifying the most constrained or "most critical" decision and for specifying an initial variable ordering heuristic. We define the ordering decision with the **overall minimum slack**, to be the decision $ordering_{i,j}$ for which

$$\min\{slack(i \rightarrow j), slack(j \rightarrow i)\} =$$
$$\min_{(u,v)}\{\min\{slack(u \rightarrow v), slack(v \rightarrow u)\}\}$$

for all unassigned $ordering_{u,v}$. Using this notion of criticality, we define a variable ordering heuristic that selects this decision at each unresolved state of the search.

With respect to the decision of which sequencing constraint to post (i.e., value assignment), we intuitively prefer the decision that leaves the search with the most degrees of freedom. Thus we post the sequencing constraint that retains the largest amount of temporal slack.

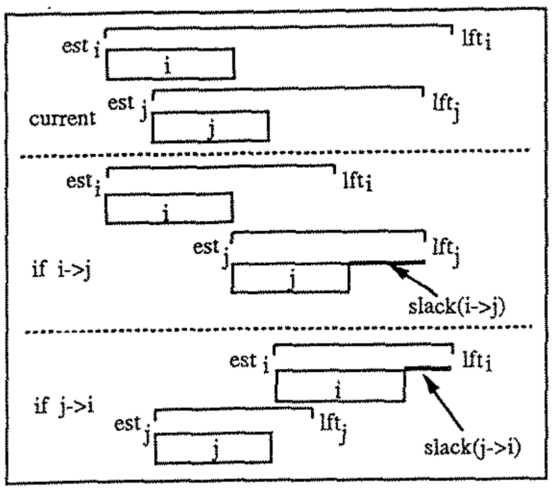

Figure 2: Slack($i \rightarrow j$) and Slack($j \rightarrow i$)

Summarizing then, our initial configuration of variable and value ordering heuristics is defined as follows:

I. **Min-Slack** variable ordering: Select the sequencing decision with the overall minimum temporal slack. Suppose this decision is $ordering_{i,j}$.

II. **Max-Slack** value ordering: choose the sequencing constraint $i \rightarrow j$ if $slack(i \rightarrow j) > slack(j \rightarrow i)$; otherwise choose $j \rightarrow i$.

A Computational Study

In this section we evaluate the performance of the above heuristics in conjunction with the PCP search procedure on a suite of job shop scheduling CSPs studied by two recently developed scheduling procedures: ORR/FSS (Sadeh 1991) and CPS (Muscettola 1993). Both ORR/FSS and CPS rely on global estimations of resource contention to dynamically direct their respective search processes. In the former case, profiles of resource demand over time are deterministically constructed according to probabilistic assumptions, and inspected to identify contention "peaks". In the case of CPS, expected resource "conflicts" are identified from demand profiles constructed via stochastic simulation in a relaxed solution space where resource constraints are ignored. ORR/FSS and CPS also differ in the type of decision that is taken at each step of the search. ORR/FSS takes a decision to fix the start time of the operation contributing most to the highest contention peak. CPS identifies the set of operations involved in the most severe resource conflict, and posts a sequencing constraint to reduce the level of contention among these operations. Unlike our approach, which establishes orderings between pairs of operations, CPS posts precedence constraints between sets of operations and attempts to post only as many constraints as are necessary to eliminate the possibility of resource contention (thus retaining additional flexibility in the final solution). Both ORR/FSS and CPS have reported very strong results on the set of scheduling problems used in this study.

As an additional point of comparison, we also include results obtained with three priority dispatch rules from the field of Operations Research: EDD, COVERT, and ATC (Vepsalainen & Morton 1987). These heuristics are frequently used and have been determined to work very well in job shop scheduling circumstances where expected job tardiness is low (as would likely be the case if a feasible solution exists).

The set of problems used in this study come from the dissertation of Sadeh (Sadeh 1991). The problem set consists of 60 randomly generated scheduling problems. Each problem contains 10 jobs and 5 resources. Each job has 5 operations. In all problems, deadlines were generated randomly within a specified range. A controlling parameter was used to generate problems in three different deadline ranges: wide (w), median (m), and tight (t). A second parameter was used to generate problems with both 1 and 2 "bottleneck" resources. Combining these two parameters, 6 different categories of scheduling problems were defined, and 10 problems were generated for each category. The problem categories were carefully defined to cover a variety of manufacturing scheduling circumstances. While each problem has at least one feasible solution, they range in difficulty from easy to hard.

The results obtained on these problems, along with those previously reported, are given in Table 1 (where problem difficulty increases from top to bottom). The number of problems solved by each approach by problem category are indicated. In the case of ORR/FSS runs, search was terminated on a given problem if a solution was not found after a pre-determined number of search states had been expanded. Two sets of results are reported for this procedure, Sadeh's original dissertation results using simple chronological backtracking and a subsequent study (labeled ORR/FSS+) where the original procedure was augmented with the "intelligent" backtracking techniques described in (Xiong et al. 1992). In the case of CPS, which operates with a stochastic resource analysis, the search was restarted from scratch upon detection of an infeasible state. In the case of our approach, no backtracking mechanism was employed and the search was terminated in failure if an infeasible solution state was reached.

Examining the results, we see that our simple slack-based variable and value ordering heuristics, in conjunction with the search space pruning techniques provided by CBA, perform remarkably well in comparison to both contention-based scheduling procedures, and while not solving all 60 problems, provide evidence in support of our hypothesis that comparable performance can be obtained with localized and less sophisticated look-ahead analysis techniques. From the standpoint of computational performance, average solution times of 128 and 78 seconds were obtained

	PCP	ORR FSS	ORR FSS+	CPS	Edd	Co-vert	Atc
w/1	10	10	10	10	10	7	7
w/2	10	10	10	10	10	8	8
m/1	10	8	10	10	8	5	5
m/2	10	9	10	10	8	8	8
t/1	10	7	10	10	3	6	6
t/2	6	8	10	10	8	8	8
sum	56	52	60	60	47	42	42

Table 1: Results of the experiments

with ORR/FSS+ and CPS respectively in these experiments. Our approach averaged 0.2 seconds for each solved problem.[2]

In the next section we attempt to refine our initial variable and value ordering heuristics to improve problem solving performance. We note in passing that the priority rules perform rather poorly on this set of scheduling problems.

Incorporating Additional Search Bias

While **Min-Slack** performed quite well over the tested problem set, it does not in fact utilize all of the information provided by the temporal slack data. In particular, it relies exclusively on the smaller slack value in determining the criticality of a ordering decision $ordering_{i,j}$, and ignores any information that might be provided by the larger one.

The most common problem created by disregarding this additional value appears in a form of tie-breaking. Consider the following example. Suppose that we have two unsequenced operation pairs, one with associated temporal slack values of $(20, 3)$, and the other with values of $(4, 3)$. **Min-Slack** does not distinguish between the criticality of these two ordering decisions, since the minimum value in both cases is 3. In the event that the overall minimum slack over all candidate decisions is also 3, then **Min-Slack** will choose randomly. But, in this case sequencing the second operation pair is certainly more critical since the flexibility that will be left after the decision is made will be considerably less than the flexibility that will remain if the first unsequenced operation pair is instead chosen and sequenced.

Given this insight, we define a second variable ordering heuristic, which operates exactly as **Min-Slack** except in situations where more than one pending decision $ordering_{i,j}$ is identified as a decision with overall minimum temporal slack. In these situations, ties are broken by selecting the decision with the minimum larger temporal slack value. Applying the PCP procedure with this extended heuristic to the same suite of 60 problems yielded 57 solved problems. Although this improvement is slight, it suggests the potential advantage of incorporating additional information.

A more subtle problem created by the information ignorance inherent in **Min-Slack** is the problem of similarity. Let's again consider an example. Suppose that we are again deciding between two unsequenced operations pairs. This time the temporal slack values associated with the first are $(3, 3)$, and the values associated with the second are $(5, 5)$. Which one is more critical? Without any ambiguity, the first one is more critical than the second one, and this is also the answer provided by **Min-Slack**. But what if we change the values for the first pair to $(20, 3)$. Is the first pair still more critical than the second one? The answer is not obvious. The point is that there exists a tradeoff between relying on minimum slack values and relying on information relating to the degree of similarity of both slack values in determining criticality. The strong performance of **Min-Slack** suggests that minimum slack values should remain the dominant consideration. But we hypothesize that the introduction of bias to increase criticality as the similarity of large and small slack values increases and decrease criticality as the slack values become more dissimilar might provide more effective search guidance.

Let us define a measure of similarity in the range [0, 1] such that for slack value pairs with identical values, the similarity value is 1 and as the distance between large and small slack values increases, the similarity value approaches 0. More precisely, we estimate the similarity between two slack values by the following ratio expression:

$$S = \frac{min\{slack(i \rightarrow j), slack(j \rightarrow i)\}}{max\{slack(i \rightarrow j), slack(j \rightarrow i)\}} \qquad (5)$$

Given the definition of S and the direction of bias desired, we now define a new criticality metric, referred to as biased temporal slack, as follows:

$$Bslack(i \rightarrow j) = \frac{slack(i \rightarrow j)}{f(S)}, \qquad (6)$$

where f is a monotonically increasing function.

With little intuition as to the appropriate level of bias to exert on the criticality calculation, but assuming that the level of bias should not be too great, we use $\sqrt[n]{S}, n \geq 2$, to define a set of alternatives, yielding

$$Bslack(i \rightarrow j) = \frac{slack(i \rightarrow j)}{\sqrt[n]{S}}. \qquad (7)$$

By empirical reasoning, we also define a composite form of the metric with two different parameters, n_1 and n_2, as

$$Bslack(i \rightarrow j) = \frac{slack(i \rightarrow j)}{\sqrt[n_1]{S}} + \frac{slack(i \rightarrow j)}{\sqrt[n_2]{S}}. \qquad (8)$$

Table 2 presents results obtained using **overall minimum Bslack** as a variable ordering criterion for different values of n in Eqn. (7) and n_1 and n_2 in Eqn. (8)

[2]All computation times were obtained on a Decstation 5000. Both ORR/FSS and CPS are Lisp-based systems; our procedure is implemented in C.

	$n=2$	$n=3$	$n=4$	$n_1=2$ $n_2=3$	$n_1=3$ $n_2=4$
w/1	10	10	10	10	10
w/2	10	10	10	10	10
m/1	10	10	10	10	10
m/2	10	10	10	10	10
t/1	10	10	10	10	10
t/2	8	8	8	10	9
total	58	58	58	60	59

Table 2: Performance using Min-Bslack heuristic

on the same suite of 60 problems. From the results, we can see that use of $Bslack(i \rightarrow j)$ as a variable ordering criterion does in fact yield improved performance on this suite of 60 problems. As expected, performance is sensitive to the amount of bias specified. In the case where all 60 problems are solved, average solution time was 0.3 seconds.

Conclusions

In this paper, we have proposed and evaluated new heuristics for solving the job shop scheduling problem with non-relaxable time windows. Our hypothesis has been that by approaching the problem as one of establishing sequencing constraints between pairs of operations requiring the same resource and by exploiting analysis techniques for limiting the search of possible sequencing decisions, simple, localized look-ahead techniques can yield problem solving performance comparable to techniques that rely on more sophisticated analysis of resource contention. We defined a series of attention focusing heuristics based on simple analysis of the temporal flexibility associated with different sequencing decisions, and a similarly motivated heuristic for determining how to sequence a given operation pair. Evaluation of these heuristics on a suite of benchmark problems previously investigated by two contention-based scheduling procedures has shown that our heuristics provide comparable results at very low computational expense.

Our current interest is in adapting the PCP approach to solve more common, optimization-based formulations of scheduling problems. In this context, certain problem constraints (e.g., due dates) are not interpreted as rigid, but instead specify preferred values over which objective criteria are defined (e.g., minimizing tardiness cost). To adapt the PCP procedure to this class of problems, two basic issues must be addressed. First, since CBA depends on the assumption that time and capacity constraints are non-relaxable, its advantage as a search space pruning mechanism is lost. We are exploring use of an alternative mechanism, inspired by standard branch and bound search procedures, which bases pruning on a dynamically refined upper bound solution. The second issue concerns the inappropriateness of temporal slack as a basis for estimating the criticality of various ordering decisions. This metric, however, can be straightforwardly replaced by the objective function itself (e.g., computing the increase in tardiness cost resulting from alternative ordering decisions for a given pair of operations), giving rise to variants of the variable and value ordering heuristics defined in this paper.

References

Bensana, E., Bel, G., and Dubois, D. 1988. OPAL: A multi-knowledge based system for job-shop scheduling. Int. J. Production Research, 26(5), 795-819.

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. Characterizing the set of feasible sequences for n jobs to be carried out on a single machine. *European Journal of Operational Research*, 4, 189 - 194.

Garey, M. R. and Johnson, D. S. 1979. *Computers and Intractability, a Guide to the Theory of NP-Completeness*, W.H. Freeman Company.

Johnston, M. D. 1990. SPIKE: AI scheduling for NASA's Hubble Space Telescope. In *Proc. 6th IEEE Conference on AI Applications, Santa Barbara, CA.*

Keng, N. and Yun, D. Y. Y. 1989. A planning/scheduling methodology for the constrained resource problem. In *Proc. IJCAI-89*, Detroit, MI.

Kerr, R. M. and Walker, R. N. 1989. A job shop scheduling system based on fuzzy arithmetic. In *Proc. 3rd Int. Conf. on Expert Systems and the Leading Edge in Production and Operations Management.* M.D. Oliff, Ed. 433-450, Hilton Head Island, SC.

Minton, S., Johnston, M. D., Philips, A. B., and Laird, P. 1990. Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair method. In *Proc. AAAI-90*, Boston, MA.

Muscettola, N. 1993. Scheduling by Iterative Partition of Bottleneck Conflicts. In *Proc. 9th IEEE Conference on AI Applications*, Orlando, FL.

Sadeh, N. 1991. Look-ahead Techniques for Micro-Opportunistic Job Shop Scheduling. CMU-CS-91-102, School of Comp. Sci., Carnegie Mellon Univ.

Vepsalainen, A. P. J. and Morton, T. E. 1987. Priority rules for job shops with weighted tardiness costs. *Management Science*, 33(8), August, 1035-1047.

Xiong, Y., Sadeh, N, and Sycara, K. 1992. Intelligent Backtracking Techniques for Job Shop Scheduling. In *Proc. 3rd Int. Conf. on Principles of Knowledge Representation*, Cambridge, MA.

Zweben, M., Deale, M., and Gargan, R. 1990. Anytime Rescheduling. In *Proc. DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, Morgan Kaufmann Pub.