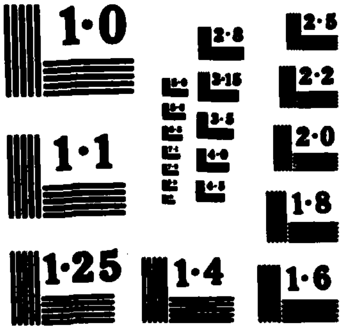
AD-A145 613 CONSTRAINT-BASED SCHEDULING IN AN INTELLIGENT LOGISTICS 1/1
SUPPORT SYSTEM: A..(U) CARNEGIE-MELLON UNIV PITTSBURGH
PA ROBOTICS INST M S FOX ET AL. 21 JUL 83
UNCLASSIFIED AFOSR-TR-84-0828 F49620-82-K-0017 F/G 5/1 NL

END
FILMED
DTIC


1·0
2·8
2·5
3·2
2·2
3·6
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

REPORT DOCUMENTATION PAGE

4

AD-A145 613

1b RESTRICTIVE MARKINGS

3 DISTRIBUTION AVAILABILITY OF REPORT
approved for public release; distribution unlimited.

DECLASSIFICATION / DOWNGRADING SCHEDULE

PERFORMING ORGANIZATION REPORT NUMBER(S)

5 MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 84- 828

NAME OF PERFORMING ORGANIZATION
Carnegie-Mellon University

6b OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
Air Force Office of Scientific Research

ADDRESS (City, State, and ZIP Code)
Intelligent Systems Laboratory
Robotics Institute, Pittsburgh PA 15213

7b. ADDRESS (City, State, and ZIP Code)
Directorate of Mathematical & Information
Sciences, AFOSR, Bolling AFB DC 20332

NAME OF FUNDING/SPONSORING ORGANIZATION
AFOSR

8b. OFFICE SYMBOL (If applicable)
NM

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F49620-82-K-0017

ADDRESS (City, State, and ZIP Code)
Bolling AFB DC 20332

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61102F	2304	A2	

1. TITLE (Include Security Classification)
CONSTRAINT-BASED SCHEDULING IN AN INTELLIGENT LOGISTICS SUPPORT SYSTEM: AN ARTIFICIAL INTELLIGENCE APPROACH

2. PERSONAL AUTHOR(S)
Mark S. Fox and Stephen F. Smith

a. TYPE OF REPORT
Interim

13b. TIME COVERED
FROM 15/3/83 TO 14/3/84

14. DATE OF REPORT (Year, Month, Day)
Jul 83

15. PAGE COUNT
15

SUPPLEMENTARY NOTATION

COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

ABSTRACT (Continue on reverse if necessary and identify by block number)

This report summarizes the progress of research performed under AFOSR Contract F49620-82-K-0017, titled "Constraint-Based Scheduling in an Intelligent Logistics Support System: An Artificial Intelligence Approach." During the contract period from March 1982 to March 1984, A theory of hierarchical, opportunistic constraint-directed reasoning for the scheduling of job shops has been the focus of the research. In addition, new research in the areas of constraint-directed diagnosis, and reactive scheduling was initiated. An experimental software system, called ISIS, has continued its evolution and has been tested on simulated plant data.

0 DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

2a NAME OF RESPONSIBLE INDIVIDUAL
Dr. Robert N. Buchal

22b TELEPHONE (Include Area Code)
(202) 767- 4939

22c OFFICE SYMBOL
NM

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Constraint-Based Scheduling in an Intelligent Logistics Support System: An Artificial Intelligence Approach

Annual Report

15 March 1983 - 14 March 1984

AFOSR Contract Number F49620-82-K-0017

Mark S. Fox
Stephen F. Smith

Intelligent Systems Laboratory
Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

21 July 1983

Abstract

This report summarizes the progress of research performed under AFOSR Contract Number F49620-82-K-0017, titled "Constraint-Based Scheduling in an Intelligent Logistics Support System: An Artificial Intelligence Approach". During the contract period from march 1982 to march 1984, a theory of hierarchical, opportunistic constraint-directed reasoning for the scheduling of job shops has been the focus of our research. In addition, new research in the areas of constraint-directed diagnosis, and reactive scheduling was initiated. An experimental software system, called ISIS, has continued its evolution and has been tested on simulated plant data.

Table of Contents

1. Introduction 1

2. Proposed Research Overview 1

2.1. Constraint Classification 2

2.2. Constraint Resolution 3

2.3. Constraint Relaxation 4

2.4. Hierarchical-Opportunistic Scheduling 5

2.5. Integrating Management Science Results 7

2.6. System Application 8

2.7. Comparative Analysis 8

3. New Areas of Research 9

3.1. Constraint-Directed Diagnosis 9

3.2. Organization Modeling 9

3.3. Reactive Scheduling 10

4. Observations and Conclusions 10

5. Publications 12

6. Research Staff 13

7. Theses 13

8. Related Presentations: March 1983-March 1984 14

Research Objectives and Status

1. Introduction

This report summarizes the progress of research performed under AFOSR Contract Number F49620-82-K-0017, titled "Constraint-Based Scheduling in an Intelligent Logistics Support System: An Artificial Intelligence Approach". A theory of hierarchical constraint-directed reasoning for the scheduling of job shops has been the focus of our research. An experimental software system, called ISIS has been constructed and tested on simulated plant data. During the period of march 1983 to march 1984, the following tasks were performed:

- The entire ISIS system was re-implemented in a more powerful version of the SRL knowledge representation system. This re-implementation effort required a significant amount of effort.

- ISIS's search architecture was further generalized to support opportunistic search and constraint-directed diagnosis.

- Research into constraint-directed diagnosis was initiated and an algorithm designed.

- Research into reactive scheduling was initiated resulting in an implementation.

- Research into OR scheduling algorithms continued resulting in new theories for the scheduling of proportional flow shops.

A complete description of ISIS can be found in Fox (1983). In the following report, the goals of the original proposal are reviewed, and our progress towards them described. It is recommended that the above technical report be read before proceeding.

2. Proposed Research Overview

We propose to construct an intelligent scheduling system for logistics support. Our research will take an eclectic approach, combining artificial intelligence (AI) and operations management (OM) research. We will investigate the application of AI constraint-analysis and hierarchical-opportunistic reasoning strategies, with OM dispatch scheduling heuristics, to design and construct an intelligent, interactive scheduling system that is applicable to domains of interest to AFOSR. In particular we will investigate the issues of:

- identifying the variety of constraints that exist in two or more scheduling applications. The examination will result in a classification of ***constraint types*** and a constraint representation scheme.

- the construction of a theory of ***constraint resolution***. There are two aspects to the resolution problem. The first is concerned with determining what constraints impact a particular decision, and the second is concerned with how to resolve conflicts among two or more constraints.

- improving schedules by allowing ***constraint relaxation***. In order to resolve conflicts among constraints, the only solution may be to re-examine the constraints to ascertain those of lesser importance, and relax them.

- reducing scheduling time by altering the system architecture to use hierarchical-opportunistic reasoning.

- expanding and integrating management science scheduling results into the reasoning process. Dispatch rules are good heuristics and can be integrated in the search process.

- doing a comparative analysis of systems and constraints.

2.1. Constraint Classification

The original proposal stated that:

> Because of the relative youth of the constraint analysis field, much work remains in the area of cataloging the constraint space. Before any useful theory is developed, the phenomena must be understood. REF-ARF dealt with constraints on values of variables in linear equations. MOLGEN also dealt with value constraints, though lisp expressions were used to specify the constraints. Each specified a predicate that the value must satisfy, and hence, can be called a *predicate (binary) value constraint*. In the scheduling domain, the *due date* of a task is a *predicate variable constraint*; it specifies one or more acceptable finish dates. The compatibiiity rules of-relaxation, because of their continuous rating assignment, can be called a *preferential value constraint*.
>
> The same dichotomy can be applied to the selection of operators during search. Either an operator is usable or not (*predicate operator constraint*), or they are ordered according to preference (*preferential operator constraint*). STRIP-like operators define their applicability constraint in the pre-condition, while NUDGE uses rules to specify preferences among different operators for relaxing constraints.
>
> Another type of constraint, commonly found in scheduling applications, is a *sequencing constraint*. When constructing a schedule for assembling a part or doing maintenance on an object, there are operations that must be sequenced in time. A *time sequence constraint* specifies the *relative* time that operations are to be completed. For example, in a turbine, a row of blades takes two weeks to install, hence each row should be produced two weeks apart, and the first row produced provides the absolute date constraint. Operations or states may also have sequencing constraints. Though it is usually the task of the planning module to determine the ordering of states and operations, it is often useful to have explicit constraints on their sequencing either stored in the system or provided by a user. The latter capability can be found in IMS and R1 (McDermott & Steele, 1981).
>
> We propose to analyze two different real-world applications to identify, categorize, and represent their scheduling constraints. This is necessary in order to design a general constraint-based scheduling system that can be applied to more than one domain.

An indepth analysis of the types of knowledge to be represented by constraints, and the types of knowledge to be represented in a constraint was performed. The types of knowledge represented by constraints include: organizational goals, preferences for resources or activities, availability of resources, and enabling states for activities. Constraints in ISIS are represented as schemata in the SRL/1.5 knowledge representation language (Wright et al., 1984). A constraint schema contains the following information:

Specification: There is an explicit specification of constraints as schemata in the representation, attached as meta-information to the domain model

Relaxation: Alternative values for the satisfaction of constraints are specified explicitly in the constraint schema.

Context: The contextual applicability of a constraint is implicitly specified by its attachment to the affected portion of the domain model, and further specified by explicit knowledge.

Importance: All constraints do not exert the same influence, hence the representation specifies the relative importance of satisfying various constraints.

Interdependence: The effect of satisfying one constraint on the satisfaction of another is required in order to diagnose and repair poor schedules. Interdependence is represented explicitly.

Utility: The utility of a value for a constraint to the rating of a schedule is also represented explicitly.

Constraints form a type hierarchy differentiating between discrete and continuous constraints, and subtypes within each of these. Constraints may be attached to schemata, slots and/or values at the meta-level with in SRL/1.5. Fox (1983) contains a complete description of the above.

The concept of time was also found to play an important role in the representation of constraints. In particular, the concept of a time varying constraint for the representation of shifts and other time dependent constraints was introduced. Smith (1983) elaborates this further.

2.2. Constraint Resolution

The original proposal stated:

> Constraint resolution is concerned with determining what constraints apply to the current decision. In planning systems like GPS and STRIPS, the constraint is the initial and goal statement, and the operators' pre- and post-conditions define further constraints depending on whether the planning is forward or backward. As MOLGEN develops a plan, it formulates new constraints from operator conditions and propagates them to other sub-problems. Hearsay-II and ABSTRIPS define *a priori* the levels of representation, which in turn define the constraints applied at each level. In all cases, there is a predefined set of constraints from which a subset is applied depending on where planning has led.
>
> In our analysis of a job-shop, we found that there are a large number of constraints of

differing variety that originate from various parts of the organization. Much of the scheduler's task is determining *what* these constraints are, and their *relevance* to the particular scheduling problem. For example, the advance planning department constructs multi-year plans which determine the production for a facility. The plan acts as a constraint on facilities used and the level of capacity to utilize them. Now under normal conditions, say producing orders with long lead times, the capacity constraint will be followed and the facility scheduled to capacity. But under abnormal circumstances, say a rush order, the constraint is in conflict with another constraint, the order's due date. The system must resolve which constraint takes precedence in this situation.

This problem of resolving conflicting constraints is dependent upon the *criteria* used to measure the goodness of a schedule. If minimizing tardiness is most important, the objective is to schedule to meet as many due dates as possible, leaving utilization and throughput as secondary concerns. If maximizing utilization of machines and labor is most important, the objective is to spread the load as much as possible to keep everything and everyone busy, even at the expense of due dates and throughput. If maximizing productivity is most important, the objective is to maximize the number of orders produced in a given time period, even at the expense of due dates, and utilization. Now, maximizing utilization usually amounts to maximizing throughput, so there is not much of a conflict there. However, there is often a conflict between minimizing tardiness and maximizing throughput.

To solve this problem, we propose a two step approach. First, alternative scheduling criteria will be modeled. Each criterion will denote the relative importance of constraints. The second, much more difficult step, is to add another layer of reasoning to the scheduling system, which when given a scheduling criterion, infers the relative importance of constraints.

The first step of the approach has been implemented and limited testing performed. Classes of orders/goals have a schema description which defines the relative importance of constraints. In addition, each constraint has an importance metric assigned to it. See Fox et al. (1982) and Fox (1983) for a discussion of scheduling goals.

2.3. Constraint Relaxation

The original proposal stated:

The reaction of planning-constraint systems to conflicting constraints is to either look for a different plan, query the user for alterations, or end processing. In the CONSTRAINTS system for circuit analysis, conflict resulted in a null value set. The system would determine what assumptions are in conflict and ask the user to resolve them. In MOLGEN, an alternative plan is searched for when value constraints cannot be satisfied. In these two systems, the constraints are either satisfied, or they are not satisfied, there is no middle ground.

In the scheduling domain there is a middle ground. As the NUDGE system showed, constraints can be preferentially ordered. In job-shop scheduling, due dates can be missed by small amounts. more labor hired in overtime if there are not enough machine operators, orders sub-contracted, costs increased or reduced, etc. That is, constraints

> can be relaxed from their original definition. Consider the specification of a due-date constraint. While meeting the due-date is important, shipping a bit early or a bit late is also ok, but being too late (tardy) or too early is not reasonable.
>
> Our research will explore the representation of relaxation constraints and how to use them to resolve conflicts.

The representation of constraint relaxation covers both predicate and choice relaxations over values, slots and schemata (Fox et al., 1983; Fox, 1983). In addition, the concept of relaxation has been extended to include time as a major component so that alternatives may be represented over time (e.g., varying shift specifications over time) (Smith, 1983). Lastly, the representation of constraints encodes the concept of interaction amongst constraints, enabling ISIS to determine how the binding of one constraint affects others, and, conversely, how the relaxation of one constraint may affect another (Fox et al., 1983; Fox, 1983).

ISIS performs a hierarchical search in which constraints are incorporated in both the generation of states and their testing. The following types of relaxation are performed during search:

Generative: For each constraint an operator may be defined and placed in the constraint's generator slot. The operator operationalizes the constraint by generating successors to a search state which contain values which represent relaxations of the constraint. For example, a next-activity constraint specifies what the next activity should be for an existing activity, but also specifies any relaxations. The corresponding operator will generate a separate successor state in the search for a schedule for each alternative activity (i.e., operation).

Analytic: After a schedule is produced, the final utility of the schedule is determined to ascertain whether the schedule is acceptable. If the utility of the schedule does not exceed a threshold, then a set of rules determines which constraints were both important but did not have a high enough utility. The search is then performed again with the importance and/or utility of these constraints altered to place greater emphasis on them.

Fox (1983) contains a detailed description of this process.

2.4. Hierarchical-Opportunistic Scheduling

The original proposal stated:

> One of the biggest problems in tackling real applications is the size of the search space. Hierarchical and opportunistic reasoning processes have contributed significantly to the reduction of this problem. Though both processes have been applied to the planning domain, none have appeared in scheduling. Consider the size of the search space for finding a schedule for 10 orders that require five machining operations (a planning module may have postulated the five operation sequence). Or ten aircraft going through five maintenance operations. On each machine, orders can be sequenced in 10! ways, hence the search space contains $(10!)^5$ solutions of which only a few satisfy the scheduling criteria and constraints. In a real job-shop, the number of orders run into the hundreds if

not thousands, and the number of operations in the tens. The space has to be reduced.

To do hierarchical scheduling, resource contention-describing abstractions must be found. For example, machines can be grouped into facilities, and a capacity-utilization-over-time metric used to describe the facility's availability. Where Hearsay-II, ABSTRIPS, and MOLGEN were a single dimension layering of abstractions, i.e., chain of abstractions as opposed to a network, scheduling requires multiple, orthogonal hierarchies. Each hierarchy describes multiple abstractions of a resource being contended for.

To do opportunistic scheduling, the system must recognize the most important and most certain constraints, at any level, and in any hierarchy, and use these as islands from which to expand the schedule.

We propose to investigate the set of abstractions that best describe scheduling and reduce the combinatorics of search. A system will be constructed, modeled after the Hearsay-II system. The BB will be multidimensional, reflecting time and abstraction spaces for each resource. Knowledge sources will be constructed that incorporate knowledge to map and extend states in the abstraction spaces. Special attention will be devoted to seeking constraints that provide "islands of certainty".

The first part of our goal has been achieved. A hierarchical reasoning system has been constructed which performs four levels of analysis in order to construct a schedule (Fox et al., 1983; Fox, 1983). A significant improvement of the schedules have been found when compared to non-hierarchical search.

We were unable to investigate opportunistic reasoning strategies to the extent that we had hoped, due to the amount of time it took to recode the system in a newer version of SRL. Nevertheless, the following issues were investigated:

Focus of attention: The first issue in developing an opportunistic search paradigm is the development of focus of attention heuristics. Two approaches are under continued investigation. The first approach uses techniques for the detection of bottlenecks. Bottlenecks form "islands of certainty" to the extent that they should be scheduled first. The OR portion of our research has played a significant role in the development of algorithms for detecting bottlenecks. In particular, the techniques defined in Ow (1984) are used to construct a preliminary schedule in which bottlenecks are easily measured.

The second approach has been the use of constraint interactions in order to determine at what level in the search hierarchy problems lie and where alterations should be made. See the section of constraint directed diagnosis for further information (section 3.2).

System Architecture: The representation of ISIS's search architecture has been altered to be more schema based. That is, each level, and each phase of search is represented as schemata, and an explicit search manager has been introduced to operate with this representation. This enables ISIS to reason about the specific levels searched and act to alter the the global search stıategy when appropriate.

Time Propagation: One source of certainty in the construction of schedules is the existence of time constraints which specify the time of execution for particular operations. Such constraints may be generated either by a higher level in the search space or by the user. In either case, it is necessary to determine what the affect will be on the rest of the schedule both forward and backwards from the specific operation. A theory and implementation for time propagation was constructed. A detailed description may be found in Smith (1983).

Research in opportunistic reasoning is continuing during the renewal year.

2.5. Integrating Management Science Results

The original proposal stated:

> To ignore the results of management science research would be foolhardy. Many of the dispatch rules proposed to date can be incorporated in the search process. For example, the shortest processing time rule (SPT) for ordering jobs in queue can be a constraint expressing preferences on queued orders for tasks.
>
> We propose to investigate dispatch rule based scheduling to extend rules to consider more constraints. Our research to date has already yielded results in elastic due-date specification[1]. These results will be incorporated in the scheduling system.

To date, our operations management research has focused on myopic heuristics for the single machines and flow shops (Morton & Rachamadugu, 1982; Rachamadugu et al., 1982). Some of these results have been integrated into ISIS in the form of due date constraints. We have also begun investigating the use of a dynamic priority rule developed by Ow (1984) as the basis for order prioritization within ISIS. A summary of her thesis follows:

> The aim of this thesis is to study heuristic search methods for two scheduling problems: (i) the single machine problem where jobs have both early and tardy costs (ii) the weighted tardiness problem in proportinate flowshops.
>
> Each problem was studied to discover properties of optimal and locally optimal solutions that could be used as a basis of heuristics. These heuristics would then be used to guide the search for a solution to these problems. In the case of the single machine problem, we developed a priority function, called the ET Rule, and used it in dispatch mode to construct schedules. For the flowshop problem, we developed another priority function, called the Idle Time Rule, and used it in an approach that focused on bottle- necks to build schedules. Then both these priority functions were used to guide a more complex search method called Beam Search to study the relative advantage of a more sophisticated search method using the same "knowledge". This also provided an opportunity for studying the behavior of the Beam Search method itself. Extensive computational studies were performed that showed that the dispatch method for the single machine problem and the focused approach for the flowshop problem outperformed a number of the better

[1] The lab is interdisciplinary. In addition to computer scientists, there are management scientists and engineers.

known heuristic methods and gave near optimal results for the smaller problems where the optimal solutions of lowerbounds could be determined. The Beam Search method improved the quality of the results of the dispatch method and the focused approach using fairly small beamwidths and was somewhat more reliable in its performance.

2.6. System Application

The original proposal stated:

> A primary goal of this research is to develop a *general* constraint-based scheduling system. This research will be proven successful if it can span more than one domain. Our intent is to choose two applications and spend one year on each. Taking advantage of our experience in job-shop scheduling, we will apply the new scheduling system to the job-shop problem. It will drastically reduce the time necessary to analyze the domain, and allow more time to be spent on research. In the second year, we will choose an application recommended by AFOSR.

To date we have focused our attention only on the job shop scheduling application. At present it is unlikely that we will have the time and resources to apply ISIS to another domain.

2.7. Comparative Analysis

The original proposal stated:

> The typical method of verifying an AI research project is to demonstrate a few canned examoles. Few are ever really tested. A major goal in our scheduling research is to construct a system that provides good functionality. To achieve this we propose to analyze the system in two ways.
>
> The first method of analysis is to test the architecture (i.e., hierarchical-opportunistic) against a competing architecture (i.e., heuristic search) to determine performance. We then intend to run the system while varying the constraints present to determine the contributions of constraints to the overall performance of the system.
>
> Our second method is to compare the system to alternative approaches such as the dispatch rule systems found in management science.

The following tests have been performed:

- hierarchical vs flat beam search.
- alternative constraints.

In all, twelve versions of ISIS have been tested using an actual model of the host plant, with simulated orders (Fox, 1983). The results have shown that hierarchical reasoning outperforms flat search. The results on the use of alternative constraints are less conclusive at this point. More testing is to be performed during the renewal year.

3. New Areas of Research

This section describes new areas of research which were not identified in the original proposal, but have played an important role in our research.

3.1. Constraint-Directed Diagnosis

It became clear during our analysis of ISIS test data that constraints could be used to diagnosis the results of scheduling. Two types of diagnosis were identified:

Intra-level Diagnosis: By determining the utility of each constraint for a particular schedule, it was simple to detect along which dimensions (i.e., constraints) the schedule behaved poorly. Hence, a constraint's utility function could be used to detect which constraints should receive greater attention. A set of rules were then created for the post analysis phase of scheduling to force the search to be performed again but with certain constraints altered in order for them to receive greater attention (i.e., increased importance or utility).

Inter-level Diagnosis: We also noticed that constraints were highly connected. That is, a shift constraint could negatively affect the satisfaction of a due date constraint. By representing how constraints could affect each other, we are able to determine which constraints would have to be modified (i.e., relaxed), and at what level in the search the relaxation is to take place.

Again, Fox (1983) contains more detail about this problem. We believe that constraint-directed diagnosis will play an important role in ISIS's opportunistic reasoning strategy.

3.2. Organization Modeling

Initial versions of ISIS contained a very simple representation of knowledge; basically using the SRL knowledge representation system as a simple database. As the variety of knowledge to be represented increased, and the number of functions required to access also increased, it became apparent that a formalization of the ISIS knowledge representation was required. Towards this end, a theory of manufacturing modeling was constructed. This theory incorporates the concepts of

- States and acts
- Possession
- Object and parts
- Time
- Causality
- Constraints

in order to model manufacturing situations in detail. The modeling language extends beyond the manufacturing domain to encompass almost any activity situation. Fox (1983) and Smith (1983) contain detailed descriptions of the modeling system.

3.3. Reactive Scheduling

A major issue that arose is how ISIS would react to change. Once a schedule is produced, it is not often the case that the environment would remain stable enough to carry it out. For example, a machine may break down, or a higher priority order may pre-empt resources. The ability to repair schedules then became an issue. We chose to implement a "theory of minimum change" in our approach to schedule repair. If an order's schedule was invalidated due to resources unavailability, ISIS would attempt to reschedule the order, but would first generate a set of constraints which attempted to preserve as much continuity between the invalidated schedule and the revised schedule as possible. That is, the old reservations became constraints for the new scheduling run.

4. Observations and Conclusions

In reviewing the above information, we have found that many but not all of the goals we set out were achieved during the contract period. We believe that significant progress has been made in the representation of manufacturing and constraint knowledge, and in constraint-directed hierarchical search. The reaction of expert schedulers in the plant we have worked with has been so positive that they have taken a copy of ISIS to the plant to field test. Nevertheless, much research remains to be done.

The focus of our research has now turned to more complex search strategies including:

1. The inclusion of opportunistic reasoning in search - We anticipated that in resource constrained situations, anchored search (i.e., which goes forward from the first operation, or backward from the last operation) would perform poorly. Hence, the system would have to recognize the "islands" of the plant which are highly constrained and schedule them first, then perform "island driving" using beam search.

2. The use of constraints to diagnose the cause of poor schedules and to recommend which constraints to alter and at what level in the search they are to be altered in order to generate a better schedule - Since higher levels of the search utilize less information, constraints generated at those levels may incorrectly focus the search at a lower level. Hence, the system must be able to diagnose such problems and correct them.

3. The integration of OR dispatch rules as search operators and constraints

4. Further testing of the system on a variety of shop configurations including flow and job-shops.

The principle challenge in this work is to provide a *complete* theory of constraint-directed reasoning in which constraints are used to automatically: 1) define the levels of search, 2) define the search operators, 3) define rating functions and acceptance criteria, and 4) guide diagnosis. Except for simple cases, 1 & 2 are beyond our current capabilities. Solving them will reduce the amount of adaptation of the system code required for a new application. Nevertheless, by the end of the current contract period, opportunistic reasoning combined with constraint-directed diagnosis and treatment will be operating and should substantially improve schedules produced under resource constrained situations. At the same time, OR research into the cost-based dispatch rules will provide us with better search operators and rating functions.

5. Publications

Bourne D. and M.S. Fox, (1984), "Autonomous Manufacturing", *Research Review*, Robotics Institute, Carnegie-Mellon University, Pittsburgh PA.

Fox M.S., (1983), "Constraint-Directed Search: A Case Study of Job-Shop Scheduling", (PhD Thesis), Technical Report. Robotics Institute, Carnegie-Mellon University, Pittsburgh PA.

Fox, M.S., Allen, B.P., Smith, S.F., and Strohm, G.A., (1983), "Future Knowledge-based Systems for Factory Scheduling." *Proceedings of CAM-I 12th Annual Meeting and Technical Conference*, Dallas, Texas, 1983.

Smith, S.F., (1983), "Exploiting Temporal Knowledge to Organize Constraints", Technical Report CMU-RI-TR-83-12, Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, June, 1983.

Fox M.S., B. Allen, S. Smith, and G. Strohm, (1983). "ISIS: A Constraint-Directed Search Approach to Job-Shop Scheduling", *Proceedings of the IEEE Computer Society Trends and Applications*, National Bureau of Standards, Washington DC.

6. Research Staff

- Fox, Mark
- Kriebel, Charles
- Morton, Thomas
- Ow, Peng-Si
- Rachamadugu, Ram
- Smith, Stephen
- Strohm, Gary
- Wright, Mark
- Vepsalainen, Ari

7. Theses

Fox M.S., (1983), "Constraint-Directed Search: A Case Study of Job-Shop Scheduling", (PhD Thesis), Technical Report, Robotics Institute, Carnegie-Mellon University, Pittsburgh PA.

Ow, Peng-Si, (1984), "Heuristic Knowledge and Search for Scheduling", (Ph.D. Thesis), Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA.

8. Related Presentations: March 1983-March 1984

25 May 83, "Constraint-Directed Reasoning: A Case Study of Job-Shop Scheduling", IE Conference on Trends and Applications, National Bureau of Standar Gaithersburg Maryland.

2 June 83, "Managing the Factory of the Future", Tri-Service Workshop on Manufacturing Research, Invited Talk, US Army Research Office, Leesburg Virginia.

June 10, 1983, "ISIS: A Constraint-Directed Reasoning Approach to Job Shop Scheduling", 4th Workshop on Distributed Artificial Intelligence, South Hadley, Massachusetts.

July 14, 1983, "The Intelligent Management System Project", presentation for visiting Japanese Production Management Study Team, Pittsburgh, PA.

5 September 83, "Issues in the Modeling of Organizations", Workshop on Knowledge Representation and Organizational Theory, IFIP WG8.3, Lisbon Portugal.

2 November 83, "AI Approaches to Production Planning and Scheduling", Honeywell CAD & CAM Workshop, Minneapolis MN.

Oct. 25, 1983, "Steps Toward the Paperless Factory", CMU RI Industrial Affiliates Program Workshop, Pittsburgh, PA.

Nov. 9, 1983, "Future Knowledge-Based Systems for Factory Scheduling", CAM-I Annual Meeting and Technical Conference, Dallas, Texas.

10 November 83, "Planning and Simulation in Robotics and Automation", Invited Panel, IEEE Conference on Computer Software and Applications, Chicago Illinois.

8 December 83, "Artificial Intelligence in Manufacturing Planning and Control", Invited Talk, Wharton Conference on Productivity, Technology, and Organizational Innovation, Wharton School, University of Pennsylvania.

16 January 84, "Artificial Intelligence in Manufacturing, and The ISIS JOb-Shope Scheduling System", Boeing Computer Services, Seattle WA.

17 January 84, "Constraint-Directed Scheduling", Operations Research Seminar, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA.

31 January 84, "Artificial Intelligence in Manufacturing", Technology Transfer Society, Paris France.

2 February 84, "Artificial Intelligence in Manufacturing", Technology Transfer Society, London England.

11 February 84, "A Constraint-Directed Reasoning Approach to Job-Shop Scheduling", Westinghouse AI Symposium, Pittsburgh, PA.

14 February 84, "Factory of the Future", Invited Speaker and Panel, ACM Computer Science Conference, Philadelphia PA.

15 February 84, "AI in Manufacturing", Frost & Sullivan Symposium on Automation and Productivity, New York, NY.

1 March 84, "AI in Manufacturing", Gulf Oil Co., Pittsburgh PA.

March 8, 1984, "Artificial Intelligence in Manufacturing", International Symposium, Productics and Robotics: Technical and Economic Aspects, Bordeaux, France.

March 9, 1984, "Use of Artificial Intelligence Techniques in Scheduling", tutorial course organized by ADETAA (Association pour le Developpement des Techniques d'Automatisation en Aquitaine), Bordeaux, France.

15 March 84, "Artificial Intelligence in Manufacturing", Invited Seminar, Dept. of Mechanical Engineering, Massachusetts Institute of Technology.

An indepth analysis of the types of knowledge to be represented by constraints, and the types of