The Management of Job-Shop Scheduling Constraints in TOSCA

Howard Beck

AIAI-TR-121

January 1993

Paper presented at NSF Dynamic Scheduling Workshop, Florida, Jan 1993.

This work has been undertaken with the support of Hitachi Limited.

Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

© The University of Edinburgh, 1993.

Abstract

Production management involves planning, scheduling and control. Whereas MRP-II systems have contributed greatly to production planning, detailed scheduling and control remain problematic and have proved a significant research challenge. TOSCA is a research and development programme which provides an overall framework for production planning, scheduling and control based around representations of factory resource and setup capacity.

Up to this time, work on TOSCA has concentrated on predictive scheduling focusing on (i) representations for capacity and setup constraints and objectives, (ii) methods to monitor the criticality of these constraints during scheduling, and (iii) methods to allow critical constraints to be effectively managed to enable good solutions to be achieved. Current research on TOSCA is investigating re-scheduling and methods of schedule repair following shop-floor feedback.

1 Introduction

Production management involves planning, scheduling and control. It is now generally accepted that whilst Manufacturing Resource Planning (MRP-II) systems have introduced a number of benefits in terms of overall factory communication most specifically in production and inventory planning, they leave a significant gap when it comes to detailed production scheduling and control. This 'gap' has been effectively dealt with in some manufacturing environments through the use of Kanban. Such an approach to scheduling and control has proved successful in streamlining production activities within plants with high volume and a limited range of products. Where volumes are low and the product range extensive, however, Kanban has been less successful. It is within these production environments that shop floor control relies heavily on the support of detailed capacity analysis and production scheduling. The current TOSCA system provides such a detailed scheduling capability and future developments are planned to extend the system to provide an integrated production management framework.

The TOSCA project was initiated by Hitachi's Expert Systems Group as a response to the frequent requests they received for factory scheduling systems. Hitachi have been successfully using an expert systems development toolkit to build factory scheduling systems for a number of years [3], but have been aware that scheduling methodologies and tools could potentially expedite the development of scheduling applications and also improve the quality of solutions obtained.

This paper reports work undertaken by the Artificial Intelligence Applications Institute (AIAI) aimed at the development of generic factory scheduling methods and at increasing understanding of production constraints and their management through the scheduling process. In this section, the current work is placed in the broader production management context.

1.1 Scheduling in the Context of Production Management

Production scheduling should not be viewed in isolation. The potential benefits of a good scheduler can only be fully exploited if appropriate scheduling tasks are set (production planning is effective) and once schedules have been generated, the factory is able to effectively manage their execution. It is in this context, that the TOSCA programme is being undertaken.

TOSCA is an evolving system inspired by the architecture and approach of O-Plan [5], a planning and control system developed at AIAI. O-Plan adapted the blackboard architecture of problem solving, for domain independent planning problems, by engineering an efficient *agenda-based* control architecture. O-Plan2 has continued this theme, expanding the capability of the original O-Plan system by incorporating the ability to model several 'agents' involved in planning and control and allowing for the exploitation of parallel processing platforms in the future. Up to this time, TOSCA has used a relatively simple flow of control but future work is planned to more extensively exploit the O-Plan framework for reactive repair and overall production management control.

O-Plan defines activity planning within a three-agent architecture, in which the planner communicates with job assignment and execution agents. TOSCA views overall pro-

duction management from a similar three-agent perspective in which a scheduler communicates with job assignment and execution agents. This is show in Figure 1.

Below is a description of the scenario for which the production planning, scheduling and control architecture is envisaged.

The Scenario

The scenario we aim to support is as follows:

- The user specifies the scheduling task which corresponds to the planning elements of production (*i.e.*, the acceptance of orders, the selection of lot sizes, and the making available of factory resources). This process is referred to as *job assignment*.
- The *scheduler* generates a schedule ready for execution. The scheduler has knowledge of the general capabilities of the execution system, but does not need to know about the actual activities to be executed.
- The *execution system* seeks to carry out the operations specified by the scheduler while working with a more detailed model of the execution environment than is available to the job assigner and to the scheduler.

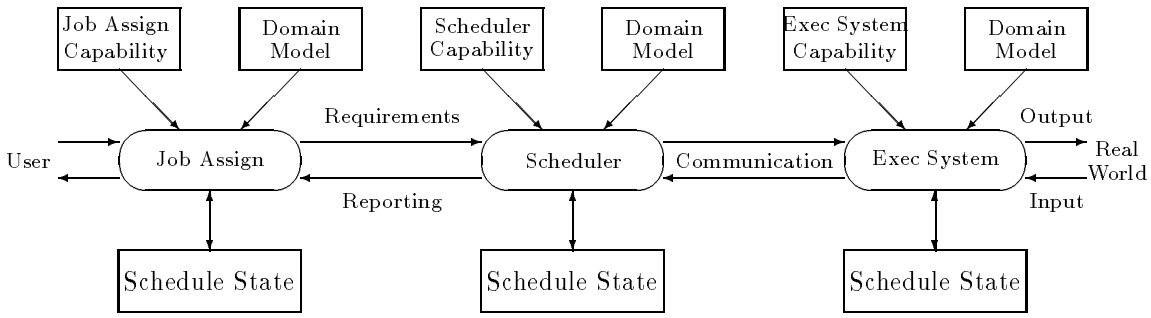

Figure 1: TOSCA three agent model: Job Assignment, Scheduler and Execution System

Communication between Agents

The agents have common representations of the capabilities of the job-assignment agent, the scheduler and execution agent, the requirements of the schedule and the schedule itself with any 'flaws' (outstanding requirements). Of particular impotance is communication relating to flaws relating to production requirements and capacity.

Based on the knowledge of its own capabilities and that of the execution environment, the scheduling agent will generate a schedule which may then be executed via an execution agent. The execution agent monitors the execution, responding to failures in one of two ways: either dealing with the problem itself or, if a repair is beyond its capabilities, feeding back a request to the scheduling agent. Where the scheduler is unable to deal with a problem, failure is fed back to the job assignment agent.

2 The Scheduling Problem

We now address the specific scheduling context to which TOSCA has been applied.

With some notable exceptions (*e.g.*, [14]), production scheduling research has concentrated on highly restricted and abstract target applications. Such abstract applications provide a means of focusing on key aspects of the job-shop scheduling problem and establish a good platform for controlled experimentation, but their validity and extendability is more difficult to establish. The TOSCA project research has attempted to achieve some level of validity by concentrating on the modelling and scheduling of real or realistic factory datasets.

The initial test domain is derived from but extends real factory data. It is based on a general job-shop but remains grounded in a genuine application. By including extensions to the real factory data, the test domain enables issues to be explored which may not exist in any currently accessible real world system.

2.1 The Test Domain

The test domain is based on a printed circuit board (PCB) fabrication covering the activities associated with the insertion of four types of components into the boards, the key characteristics of the problem being: (1) realistic scale, (2) complex constraints, and (3) JIT scheduling objectives.

Scale

- The research was based on real factory data covering more than 2 200 operations.
- The factory is able to manufacture 350 different product types. A concomitant feature was relatively frequent machine setups.
- A detailed schedule is required over a 30 day period (*i.e.* 90 eight hour shifts).
- There are 35 machines, which in terms of their processing capabilities, can be broadly divided into four major groupings. Machines within the groupings are not all identical.

Constraints

The problem involved the following constraint types:

- Temporal precedence between operations.
- Restrictions on the frequency of machine setups (*i.e.* the number in a day). This applies to both individual machine and workcentres.
- Enforced time periods between machine setups. This applies to machines in a specified workcentre.
- Resources with overlapping capabilities. In other words, non-identical resource types may be capable of processing the same operations.

- Operation durations are fixed but contingent on the resource selected. Setup durations, too, are fixed but contingent on the resource selected.

Objectives

The objectives of the schedule can be summarised as being in accordance with the JIT philosophy. Most particularly, this involves the reduction of WIP and finished-goods inventory without introducing high levels of order tardiness. The overall level of factory inventory rests on the extent to which due dates are satisfied and the timing of the order release. The objectives of the scheduler are to avoid: (i) order tardiness and (ii) the unnecessarily early release of orders into the factories; that is, produce when required but no sooner.

3 The System

3.1 The Approach

TOSCA aims to restrict the need to backtrack, through 'intelligent' decision making based on the principles of least commitment and opportunism

The principles of least commitment have been applied to various problems domains as a means of limiting backtracking (*e.g.*, [15], [5]). In the scheduling domain, TOSCA seeks to make 'low commitment' decisions based on information regarding the current schedule state. Instead of treating operation allocations as the atomic level of scheduling commitment, the system permits alternative (lower-order) decision types which can be applied to deal with the predicted problems associated with a schedule state. These decisions act to progressively restrict the domain of possible allocations.

Opportunism, the ability to dynamically re-direct the attention of the problem-solver in the light of emerging information, has been successfully applied in various scheduling systems (e.g, [13], [2], [12]). The aim is to address the most critical constraints early in the scheduling process, before resourcing options are reduced and flexibility restricted.

The approach adopted by TOSCA involves: (i) identifying potentially threatened global constraints (*i.e.* scheduling flaws) associated with the schedule state and selecting one to address, (ii) taking 'small' decisions to tackle the flaw and (iii) propagating the consequences of each decision. The system relies heavily on constraint propagation and prediction to drive decision making.

TOSCA uses four different types of decision:

- Perform two or more operations consecutively to avoid a setup. This is achieved by adding a zero duration time constraint between the end of one operation and the start on the other.

- Restrict the resourcing options of an operation. The demand on a specific resource time period is redistributed by deciding that an operation (which could legally be allocated to that resource time period) will be processed on an alternative resource.

- Restrict the start time window of an operation. The demand on a specific resource time period is redistributed by deciding that an operation (which could legally be allocated to that resource time period) will be processed within a more restricted time period.

- Start an operation at a particular point in time.

Each decision type relates to a scheduling flaw identified from predicted threats to global constraints. Decisions are taken to reduce the predicted threat to a specific constraint. In the test data under consideration, the primary global scheduling constraint types are: (i) the machine capacity limits, and (ii) the factory imposed setup restrictions. Each of these constraints is continuously monitored across the full scheduling horizon using predictions of expected demand. Threats are identified by comparing estimates of demand for capacity and demand for setups against known capacity limits and the setup restrictions.

3.2 The Representation of Constraints

The scheduling solution must respect a number of constraints, in particular temporal-capacity constraints which may be dynamically monitored throughout the scheduling process. Temporal capacity constraints refer to capacity restrictions that apply over time; for instance, total time available for processing on a resource during a day or maximum number of setups permitted at a workcentre during a shift. TOSCA monitors demand at individual resources and groups of resources. In addition to associating operations with resource time periods through their time windows, operation preferences and associated demand are also associated with resource time periods.

Temporal-capacity constraints can be monitored in terms of the degree of threat to their satisfaction, which is a function of estimated demand and available capacity over time. Estimates of demand change in accordance with decisions taken by the scheduler, and for this reason, variables which relate to demand at resource time periods are maintained. These include the list of operations which need (or may need) time at a resource during a particular period to ensure that their start and due date constraints are not violated, and the list of operations which need (or may need) to be set up at a resource during a particular period to ensure that their start and due date constraints are not violated. These temporal constraints on operations impose a demand for time and a demand for setups at the resource time period. TOSCA represents overlapping resource time periods of various durations covering the entire scheduling horizon. Operations are associated with the smallest resource time period which covers the operation time window.

Whenever the scheduler introduces an action, an operation's resourcing options or time window is restricted. Because operations are linked by temporal and temporal-capacity constraints, scheduling actions will in general introduce further operation restrictions and also alter the profile of expected demand on resources.

The term used for the temporal constraint monitoring information which is used to guide the schedule generation process is strategic knowledge. The key features of strategic knowledge are (i) the provision of a global (constraint-based) perspective on the

scheduling problem, (ii) the identification of the most critical current constraints, and (iii) the requirement for dynamic updating of the global perspective.

Specifically with respect to support for search management, strategic knowledge provides search management information; *viz.*, the basis for

- problem decomposition,
- focusing scheduling decision making, and
- choosing between alternative allocations.

Problem decomposition:

Decomposition involves the partitioning of the problem into sub-problems which can be solved more easily than the original problem. This is an approach very widely adopted in planning and scheduling (e.g, [11], [13], [5]).

Strategic knowledge is used to decompose problems in terms of constraints; thus allowing sub-problems - a subset of operations, resources and sub-interval of the overall scheduling horizon - to be defined.

Focusing scheduling decision making:

Just how problems are decomposed greatly affects the space to be searched. This conclusion is backed up by studies of Constraint Satisfaction Problems indicating that the order in which variables are chosen for instantiation is found to have a substantial impact on the complexity of backtrack search (*e.g.*, [8], [9], [10]). Heuristics preferring the most constrained variables first are usually most successful in restricting search. The benefits of opportunistic search-rearrangement has been demonstrated both empirically and analytically for a heuristic developed by Bitner and Reingold [3] which involves repeatedly focusing on the variable with the fewest remaining alternatives.

Strategic knowledge supports opportunistic problem focusing by identifying the most critical current constraints.

Choosing between alternative allocations:

During schedule generation, it is difficult to judge which of the possible alternative decisions is best. The judgement relies on a sound analysis of the current partial schedule state. This may be provided by the strategic knowledge (*e.g.*, [12], [2]). Where decisions are taken which lead to constraint violations resulting in the need to relax constraints, the overall quality of the schedule is compromised.

Strategic knowledge informs allocation decisions by identifying the elements accounting for potential constraint threats. The scheduler can use this information to reduce or manage the most critical constraints.

Representing Strategic Knowledge: Habographs

In TOSCA, novel datastructures referred to as habographs have been defined to represent strategic knowledge. Habographs consist of a number of cells corresponding to time periods across the scheduling horizon. (The size of the habograph is set by the time granularity chosen by the user.) Each cell holds information pertaining to a constraint over

a particular period, most importantly: the operations which impose a demand over the period, the available capacity, the estimated demand and the demand pressure, the ratio of demand to available capacity which is used as measure of constraint criticality. The information provided by habographs has much in common with other capacity planning lookahead mechanisms in identifying bottlenecks (*i.e.* the most critical constraints), but in addition, they provide a mechanism for identifying constraint violations [1]. This provides the means of identifying failure in partial schedules (*i.e.* identifies schedule states from which to backtrack or relax constraints). In that the operations implicated in the constraint violations are known, possible tactics for constraint relaxation can be readily identified.

A range of different types of habograph have been defined. Each specific habograph is characterised in terms of three dimensions: (i) the type of information represented (time or setup habographs), (ii) whether the information pertains to one or more than one resource (individual or aggregate resources) and (iii) whether the demand is associated with the legal limits imposed by lot start and due dates or with the temporal preferences (limits or preference habographs).

These datastructures are widely exploited in the schedule generation process: for the management of setups, the allocation of resources of overlapping capabilities, and the management of the trade-off between hard and preference temporal constraints.

3.3 The Scheduling Method

Scheduling in TOSCA occurs at three levels: pre-scheduling, high-level scheduling and low-level scheduling.

3.3.1 Pre-Scheduling

The motivation behind the pre-scheduling phase is the identification of infeasible or tight scheduling constraints. This information can be used by the scheduler itself or be passed back to the production planning components. The scheduler may or may not accept infeasible problems. During pre-scheduling, three major functions are performed: reading in of the data files, merging operations and setting up of the resource hierarchy.

Read in: Files defining the scheduling problem (*i.e.* the process plans, the lots to be produced, the machines being used and the workcentres with their associated setup constraints) together with a file defining schedule parameters including such things as habograph granularity are initially read in.

Merging operations: Scheduling problems may be defined for which setups are a critical constraint. In such cases it is necessary to restrict the overall number of setups by enforcing the contiguous processing of operations of the same type and process plan.

In merging operations, new temporal constraints are added between operations from different lots. This reduces the time windows of each of the operations and, where previously, the start and end times of the two operations were independent, they are now tied by the following relationships between their earliest start and ends, and latest start and ends:

$$\begin{array}{lcccl} es_1 + d_1 & = & ee_1 & = & es_2 \\ ls_1 + d_2 & = & le_1 & = & ls_2 \end{array}$$

When considering operations to merge there may be alternative candidates which could merge with a specified operation. Preference is given to merging operations whose time windows overlap most. Where setup demand is very high, as many operations as possible are merged.

Building the resource hierarchy: The TOSCA scheduler uses a resource hierarchy in order to represent the relationship between resources. This hierarchy, shown in Figure 2, is split into two main subtrees, one for homogeneous and one for heterogeneous resources.

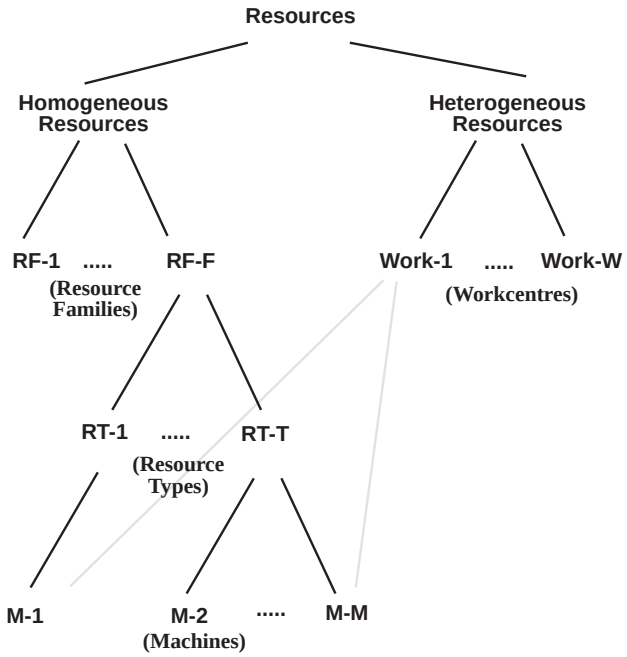

Figure 2: The Resource Hierachy in TOSCA

The homogeneous side of the hierarchy is based on abstractions of groups of resources based on their capabilities. The elements in this hierarchy from the bottom up are: resources (machines), resource types and resource families. The leaf nodes of the hierarchy are physical resources (machines). Resource types are groups of identical machines, with the identical capabilities. Resource families consist of resource types with overlapping capabilities. The heterogenous side of the hierarchy is based on abstractions of groups of resources based on mutual constraints (*e.g.*, joint restrictions on the number and timing of machine setups). These groups of resources are referred to as workcentres.

Habographs are associated both with physical resources, the leaf nodes of the resource hierarchy, and abstract resources, the non-leaf nodes on both sides of the hierarchy. Currently, habographs are not associated with resource types (*i.e.* groups of identical

resources. Time habographs are used on the homogeneous side of the tree and setup habographs are used on the heterogeneous side.

Since there is ultimately only one actual set of operations being modelled there is inevitably a close relationship between the habographs at the various level of the resource tree. In all cases when a change is made to one habograph, the results of that change need to be propagated across the resource hierarchy and into affected habographs. The habograph module within TOSCA performs this propagation automatically in order to keep the schedule in a consistent state at all times.

3.3.2 High-level Scheduling

High level scheduling involves two phases: (i) resource allocation and (ii) high level temporal allocation. The high level scheduler allocates a specific resource and narrow time period in which the processing of the operation will start and its output serves as the direct input to the low-level scheduler.

Resource allocation

Resource allocation in TOSCA is concerned with balancing demand for time and demand for setups across resources. Because demand for time and demand for setups are in general positively correlated, balancing the one tends to have the effect of improving the balance of the other. The decision as to which constraint type (time or a setup) is taken on the basis of the estimated criticality of the constraint, it being most important to ensure a good balance at the most critical resource.

The initial focus during resource allocation is set by the most critical time period at the aggregate family level (*i.e.* resource family or workcentre). This becomes the problem focus and is decomposed both in terms of its constituent resources and time periods. The task is to achieve an effective balance across the involved resources and time periods. The approach adopted is, considering time period by time period, to balance demand by deciding not to use the most heavily loaded resource for one of its possible operations. Operations associated with the narrowest time periods are dealt with first and the dropping of operation resourcing options continues until each operation has but a single resourcing option (*i.e.* resources have been allocated).

High-level temporal allocation

High-level temporal allocations are achieved by the iterative refinement of operation time windows. As is the case with dropping resourcing options, operation time window restriction decisions are driven from a capacity constraint perspective.

Resource time periods with excessive time or setup demand indicate a constraint threat which prompts the redistribution of demand by restricting the time windows of operations. Constraint criticality may be derived from either: (i) demand associated with the limits constraints (*i.e.* the outer legal limits) or (ii) demand associated with the preference constraints. The temporal preferences of JIT operations are to be processed as late as possible but still permitting the due date to be achieved. Where limits demand

pressure is above 100% (*i.e.* the demand is greater than capacity), no complete set of allocations is feasible. Where preference demand pressure is above 100% the preferred set of allocations is not feasible. Where the limits demand pressure is less than 100% and the preference demand pressure is greater than 100%, a solution may be feasible but involves a trade-off between making allocations in line with preferences (*i.e.* cost savings through inventory reduction) and maximising the number of orders meeting their due dates. Temporal restriction decisions are taken to maintain preferences where that can be done without violating due date constraints, preferences being sacrificed where necessary to avoid due date constraint violations. This relaxation of preference constraints proceeds only as far as is necessary to make the limits constraints feasible.

Temporal refinement proceeds as follows: an operation from the most critical preference time period is selected and moved from this period (*i.e.* scheduled away from its most preferred timing.) This involves restricting the operation's (limits) time window and updating its preferred timing to fall within its new limits. The operation may be selected by various criteria: the operation with the most slack (*i.e.* the operation with the largest time window duration minus the operation duration) is normally selected. This procedure of time window refinement continues until the operation's temporal preferences can be satisfied or until all operations have been restricted to the point where the limits have been restricted to the pre-defined temporal granularity.

3.3.3 Low-Level Scheduling

Following high-level scheduling, each operation has been allocated to a single resource and its start to a single time period. No specific operation start time has been set and operations allocated to the same resource and time period remain to be sequenced. This output from the high-level scheduler serves as the input to the low-level scheduler. The current system does not make provision during low-level scheduling to return to the high-level.

Two approaches have been developed for low-level scheduling. One involves a full search, and the other uses a dispatch-based approach. The feasibility of undertaking a full search depends on the number of allocation options remaining after high-level scheduling. This is primarily determined by the extent to which temporal refinement is made by the high-level scheduler.

Low-level scheduling proceeds by identifying decision points (*i.e.* start times for operations at a resource), finding all possible operation allocations, and selecting one to be allocated to this resource at this time. It also tests constraints which have not been resolved by the high-level scheduler (*e.g.*, enforcement of time periods between setups). Start times are usually identified from the predicted operation completion time (*i.e.* when the machine becomes free) but, when no operation can be allocated or when allocating an operation would result in a constraint violation, a time gap is introduced into the resource schedule. The gap is only as long as is required to ensure that no constraints are violated. In the case of the dispatch rule approach, the selection of which operation to allocate is made on the basis of a general heuristic, *i.e.* the operation with the earliest due date. Following each allocation, the time windows of dependent operations are updated, and, in addition, operation orderings are inferred where possible. An ordering of operations

can be inferred where for a pair of operations only one allocation sequence would allow the due dates to be met.

Future work to the high-level scheduler is planned to enable more refined temporal allocations to be made for the operations which are soon to be processed. This would allow the low-level schedule to undertake a comprehensive search for just these operations.

4 Conclusion

The size of the job-shop scheduling scheduling search space is too large to allow brute-force generate and test approach with backtracking. This paper describes a knowledge-based approach to guide the scheduling process thus restricting search. The information used to guide search, referred to as strategic knowledge, is derived from the dynamic monitoring of potential constraint conflicts, and is used to avoid setups as well as for resource and temporal allocations.

The information provided and maintained by habographs is of general value within production planning, scheduling and control and it is planned to exploit these datastructures in contexts other than predictive scheduling *viz.*, capacity planning and reactive scheduling. In that habographs offer a basis for understanding production requirements and constraints they could be used as a basis for determining resource and labour requirements (*i.e.* capacity planning). In that they provide insights into the implications of disturbances to an existing schedule, they could be usefully applied in a reactive context. Current research on TOSCA is investigating re-scheduling and methods of schedule repair following feedback from the shop floor.

References

[1] H.A. Beck.- Constraint Monitoring in TOSCA,Technical Report TR-118, Artificial Intelligence Applications Institute, University of Edinburgh, 1992.
[2] P. Berry - A Predictive Model for Satisfying Conflicting Objectives in Scheduling Problems, PhD Thesis, University of Strathclyde, Glasgow, 1991.
[3] J. Bitner and E.M. Reingold - Backtrack Programming Techniques, Communications of the ACM, 18 (11): 651-655, 1975
[4] K. Collyer - AI-based factory scheduling applications. In I.M. Graham and R.W. Milne, editors, Expert Systems and their Application, British Computer Society, 1991.
[5] K.W. Currie,.and A. Tate - O-Plan: the Open Planning Architecture, Artificial Intelligence, 51(1), 1991.
[6] B. Fox. - OPT: An Answer for America, Inventories and Production Magazine, 1982.
[7] C.P. Gomes and H.A. Beck - Synchronous and Asynchronous Factory Scheduling, Information Technology: Journal of the Singapore Computer Society, 1993 (To appear).
[8] R. Haralick and G. Elliot - Increasing Tree Search Efficiency for Constraint Satisfaction Problems, Artificial Intelligence, 14(3):199-219, 1980.
[9] P. Purdom and C. Brown - An Empirical Comparison of Backtracking Algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence on Pattern Analysis and Machine Intelligence, 4:309-316, 1980.

[10] P. Purdom - Search Rearrangement Backtracking and Polynomial Average Time, Artificial Intelligence, 21,1983.
[11] E. Sacerdoti - Planning in a Hierarchy of Abstraction Spaces, Artificial Intelligence, 5(2):115-135, 1974.
[12] N. Sadeh - Look-ahead Techniques for Micro-oportunistic job shop scheduling, PhD Thesis, School of Computer Science, Carnegie Mellon University, 1991.
[13] S.F. Smith - A constraint-based framework for reactive management of factory schedules. In M. Oliff, editor, .Proceedings of the International Conference on Expert Systems and the Leading Edge in Production Planning and Control, Charleston, SC, 1987 (May).
[14] S.F. Smith - The opis Framework for Modeling Manufacturing Technical Report CMU-RI-TR-89-30, Center for Integrated Manufacturing Decision Systems, The Robotics Institute, Carnegie-Mellon University, 1989 .
[15] M. Stefik - Planning with Constraints (MOLGEN: Part 1), Artificial Intelligence,16:111-139, 1981.