From: AAAI-88 Proceedings. Copyright ©1988, AAAI (www.aaai.org). All rights reserved.

Reactive Plan Revision

Peng Si Ow, Stephen F. Smith, Alfred Thiriez
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In this paper, we describe a methodology for reactively revising schedules in response to unexpected events. The approach is based on recognition of constraint conflicts in the existing schedule. Alternative scheduling actions, each offering selective advantages for conflict resolution, are available for resolving these conflicts. We present a model for action selection based on an analysis of the implications of specific schedule features with respect to the needs and opportunities available in resolving the conflicts. By matching these implications with the behavioral characteristics of different scheduling actions, it is possible to identify the most appropriate reaction in a given situation. Empirical evidence is included to validate portions of this model.[1]

1. Introduction

The need for reactive plan revision occurs when the environment changes during plan execution and parts of the plan become invalidated. The specific focus of our work is coordination of factory production, a domain where the activities of multiple agents (in this case, the operations associated with different production orders) are highly constrained by the need to share a finite set of resources, and efficient allocation of these resources over time is central to maximizing achievement of agents' goals. Unlike many reactive domains, these problem characteristics argue strongly for advance development of a schedule (plan), as this is the mechanism by which resource contention can be anticipated and its harmful effects minimized. At the same time, the schedule's utility is limited by unanticipated events on the factory floor (e.g. machine breakdowns). Effective planning in such dynamic environments requires the ability to "patch" the existing schedule as changes occur. We call this revision process *reactive scheduling*. This can be contrasted with reactive planning [Agre&Chapman 87], where no advance planning is performed.

In [Smith&Ow 85], we argued the insufficiency of a single "agent-based" problem decomposition in balancing the conflicting set of goals and preferences that govern factory production, and the utility of localized reasoning from both resource-based and order-based perspectives, focused by characteristics of current solution constraints.[2] This concept was validated in [Ow&Smith 88] relative to generating a complete schedule, wherein recognition of situations of likely resource contention was used to determine which scheduling decisions should be made from each perspective and in what order. In this paper, we extend this concept to reactive scheduling. Reactive scheduling involves (i) recognizing the conflicts that are introduced into the schedule as a result of a change in the environment, (ii) selecting a scheduling action to resolve these conflicts, and (iii) applying the action. The problem is additionally complicated by the "ripple effect" that spreads conflicts to other parts of the schedule as actions are applied and specific revisions are made. We describe the reactive scheduling methodology implemented in the OPIS scheduling system. A model is proposed for selecting actions, and some experimental results provide empirical evidence for parts of the model.

2. The OPIS Reactive Scheduling Methodology

The OPIS reactive scheduling methodology is founded on three basic principles:

1. Reaction should be focused on recognition and analysis of the constraint conflicts that are introduced into the current schedule.
2. Specific reactive problems suggest an emphasis on either an order-based or a resource-based perspective.
3. Meta-level control of reactive schedule revision must proceed opportunistically, repeatedly drawing on analyses of the current conflicts to determine which scheduling action(s) to perform next. The tightly-coupled nature of scheduling decisions makes it extremely difficult to predict the specific disruptive effects of a given scheduling action on the rest of the schedule (the ripple effect).

In the following subsections, we describe the OPIS approach to recognizing constraint conflicts, and the various order-based and resource-based methods available as reactive actions. Details of the control architecture can be found in [Smith 87].

[1]This research was sponsored in part by IBM Corporation under contract 71223046 and the CMU Robotics Institute.

[2]The importance of localized reasoning in parallel domains has also been argued in [Lansky&Fogelsong 87].

2.1. Recognition of Constraint Conflicts

Detection of constraint conflicts in the current schedule is the means by which the need for reaction is recognized within OPIS. Conflict detection is accomplished by incrementally maintaining descriptions of the current time bounds on operations and the current availability of required resources, both of which are represented at multiple levels of abstraction. Whenever changes are made to the descriptions of specific operations or resources, these new constraints are combined with model-defined constraints on production processes and resources to update the descriptions of related operations and resources. Constraint propagation in response to schedule changes can lead to detection of two types of conflicts:

- time conflicts - situations where the time bounds (i.e. scheduled or actual execution times) of two operations belonging to the same production plan are inconsistent, and
- capacity conflicts - situations where the resource requirements of a set of currently scheduled operations exceeds the available capacity of a resource over some interval of time.

This schedule maintenance subsystem is described in more detail in [LePape&Smith 87].

To determine the actual focal point around which reaction should be centered on any given problem solving cycle, some aggregation of the currently posted conflicts may be necessary. Conflict aggregation is intended to group together those individual constraint conflicts that should be simultaneously addressed, and, for purposes of this paper, is based on commonality of the resources involved in the conflict at some level of aggregation. In the case of capacity violations, this is (obviously) the resource being contended for. In the case of time conflicts, we assume that the downstream resource (as determined by the precedence constraint leading to the conflict) is the resource of interest from the standpoint of aggregation. We designate this resource as the *focal point resource*. It is important to note that this resource is itself typically an aggregate (i.e. a group of machines) with some amount of capacity.

2.2. Actions for Resolving Constraint Conflicts

Before turning attention to issues relating to conflict analysis and selection of reactive scheduling actions, we briefly describe the scheduling actions available in OPIS. Table 2-1 summarizes the behavioral characteristics of each alternative. The entries for a given action are assigned values in the range from 0 to 1 (with 0 being the lowest possible rating and 1 the highest), and these values are intended to broadly indicate the relative strengths and weaknesses of each action. In more detail, these scheduling actions include:

- *Order Scheduler (OSC)* - OSC provides a method for generating or revising scheduling decisions relative to some contiguous portion of a given order's production plan. It implements the constraint-directed heuristic search technique originally developed in the ISIS scheduling system [Fox&Smith 84]. This method is characterized by the use of a beam search to explore alternative sets of resource assignments and execution intervals, evaluating various alternatives with respect to how well the decisions satisfy relevant preferences (e.g. work-in-process time objectives, machine preferences). In invoking OSC, resource availability constraints can be made more or less visible. It can either be constrained to consider only execution intervals for which resource capacity currently exists, which we designate as the complete visibility (CV) OSC, or allowed to consider capacity allocated to lower priority orders as available, which we designate as the prioritized visibility (PV) OSC. Since the latter case admits the possibility of introducing additional capacity conflicts into the schedule (leading to "bumping" of lower priority orders), a decision to invoke the PV-OSC trades off potential additional disruption for some ability to perform resource-based optimization (hence the value 0.5 for this characteristic in Table 2-1).
- *Resource Scheduler (RSC)* - RSC provides a method for generating or revising the schedule of a designated resource (typically aggregate). The method is predicated on the assumption that contention for the resources in question is high and, thus, emphasizes efficient resource utilization. It generates scheduling decisions using an iterative dispatch-based approach, selectively employing a collection of dispatch heuristics to provide sensitivity to different preferences. Details of this approach may be found in [Ow&Smith 88]. In reactive contexts, RSC selectively applies the same strategy. It tentatively assumes that a complete new schedule will be generated forward in time from the point of the current conflict, but stops as soon as the new schedule can be consistently merged with the fragment of the old schedule that contains just those operations that have yet to be placed in the new schedule. Since the RSC places sole emphasis on resource-based optimization, the likelihood of additional disruption of the schedule due to time conflicts with downstream operations is high.
- *Right Shifter (RSH)* - The RSH implements a considerably less sophisticated reactive method which simply "pushes" the scheduled execution times of designated operations forward in time by some designated amount. Such initial shifts can introduce both time conflicts and capacity conflicts. However, these conflicts are internally resolved by propagating the shifts through resource and order schedules to the extent necessary. Thus, the RSH will not introduce any new conflicts into the overall schedule.
- *Demand Swapper (DSW)* - Demand swapping is a specialized reactive method that exchanges the remaining portion of one order's schedule with the correspondent portion of the schedule of another order of the same type so as to minimize their combined tardiness. Note that the DSW is not necessarily a conflict resolution strategy. It is more appropriately viewed as a scheduling action that improves the character of the conflict.

	RSC	CV-OSC	PV-OSC	RSH	DSW
resource-based optimization	1	0	0.5	0	0.5
order-based optimization	0	1	1	0	1
time conflict avoidance	0	1	1	1	0.5
capacity conflict avoidance	1	1	0	1	1
sequence stability	0	0	0	1	0

Table 2-1: Characteristics of Scheduling Actions

3. Reasoning About Reactions

Given the scheduling actions described above, the control problem is: How to best exploit the selective advantages of each in reactively resolving conflicts? As a first step to addressing this question it is important to consider criteria for evaluating the utility of various reactive revisions to the current schedule. We identify three:

- attendance to scheduling objectives - This concerns the "quality" of the result relative to expected factory behavior.
- amount of disruption - This concerns the extent to which reaction has been localized.
- efficiency of reaction - This concerns the speed of the reactive revision process.

In the following subsections, we focus on the control problem stated above. We first identify a set of features relevant to conflict analysis. Then, on the basis of the implications of these features in relation to the behavioral characteristics of the specific scheduling actions identified in Section 2.2, we arrive at a model for selecting appropriate reactions.

3.1. Conflict Analysis

Meta-level control decisions should exploit knowledge relating to both the continuing validity of various scheduling decisions and the flexibility of current time and capacity constraints. Our goal in this section is to identify a set of features that provides this knowledge. We begin by considering specific characteristics of the conflict itself:

- *conflict duration* - The duration of the conflict (or the maximum of the durations of the individual constraint conflicts if several have been aggregated) provides one indicator of the validity of the focal point resource's schedule. Appealing to sensitivity analysis [Bean&Birge 85, Johnson 74], if the duration is low, then it is reasonable to assume that the sequencing decisions previously made at the focal point resource remain valid. The problem lies only in the timing details. If the duration is high, however, this assumption is not valid.
- *number of operations in conflict* - If the number of conflicting operations is low, then it is reasonable to assume that most sequencing decisions relative to the focal point resource are valid. For example, if there is only a single conflict, then the operation in conflict may be the only one out of sequence. If the number is high, then sequence optimization at the focal point resource is an important concern.

The implications of these features are summarized in Table 3-1.

Duration	Number of Operations	Implication
low	----------	sequencing decisions remain valid
high	low	small sequence changes needed
high	high	sequence optimization needed

Table 3-1: Implications of Conflict Characteristics

We next consider features relating to the flexibility of current time and capacity constraints in the schedule. Here we are only interested in the local flexibility surrounding the conflict. To this end, we define the *conflict horizon*, an interval that temporally spans the conflict by some reasonable margin, to place temporal bounds on the analysis. We identify several additional features:

- *fragmentation of focal point resource's schedule* - This is a profile of amount of available capacity at the focal point resource over the conflict horizon (recall we are typically speaking of an aggregate resource). If fragmentation is low, then the focal point resource is a bottleneck, indicating that optimization of the focal point resource's schedule to achieve maximum throughput is a primary concern. If fragmentation is high, then resource-based optimization is unimportant.
- *local downstream slack* - This measure captures the local flexibility in the scheduled end times of the operations scheduled on the focal point resource. In defining this measure, we appeal to an assumption concerning characteristics of a good schedule, namely that good schedules will exhibit queue times only before bottleneck resources. Given this assumption, we define local downstream slack to be the average of the durations of the first scheduled delay due to resource unavailability encountered by each order scheduled on the focal point resource within the conflict horizon. If there are no scheduled delays in an order's schedule, then there is no local slack. If downstream slack is low, then downstream resource contention is not likely to be severe (i.e. there are no apparent downstream bottlenecks). If downstream slack is high, there is evidence of at least one downstream bottleneck and optimization of resource schedules further downstream

may be important.

- *local upstream slack* - This measures the flexibility in the scheduled start times of the operations scheduled on the focal point resource. In this case, slack can be defined in terms of the scheduled (or actual) end times of the immediately preceding operations (given the above assumption about the starting schedule). If the local upstream slack of operations requiring the focal point resource is high, then there are opportunities for resequencing on the focal point resource. It might be possible to place operations into the "holes" of available capacity that will be vacated by the conflicting operations.
- *projected lateness* - Relative to the specific operation(s) in conflict we also define a related measure of projected lateness. The projected lateness of an operation is defined in a similar manner to local downstream slack, except now we are interested in the difference between the earliest time that the order can arrive at the downstream bottleneck and its scheduled start time on that resource. If there is no downstream bottleneck, then we are interested in the difference between the earliest the order can finish and its due date. If the projected lateness of an operation is negative (i.e. the operation is still "early" relative to its current deadline), then resource-based optimization is unimportant. If the lateness is positive, then resource-based optimization is important.
- *variance in projected lateness of all operations in the conflict horizon* - This provides an indication of the opportunities for pair-wise optimization of order schedules. If the variance is high, then it may be possible to tradeoff positive and negative projected latenesses of specific orders by swapping demands.

The implications of these features relating to constraint flexibility are summarized in Table 3-2.

Measure	Value	Implication
Fragment. at focal point resource	low	resource-based optimization important
	high	resource-based optimization unimportant
Downstream slack	low	no downstream bottlenecks
	high	existence of downstream bottleneck(s)
Upstream slack	low	limited flexibility for resequencing
	high	opportunities for resequencing
Projected lateness of conflicting operations	neg.	resource-based optimization unimportant
	pos.	resource-based optimization important
Variance in projected lateness	low	no opportunities for demand swapping
	high	opportunities for demand swapping (in conjunction with + lateness)

Table 3-2: Implications of Constraint Flexibility

3.2. Selecting Scheduling Actions

In the previous section, we identified specific *features* of the current state of the schedule and indicated their implications relative to the validity of sequencing decisions, the opportunities for either order-based or resource-based optimization, and the scope of the reaction. These implications distinguish some scheduling actions as being more appropriate than others with respect to the evaluation criteria stated earlier. Thus, by consolidating these features of the current control state and analyzing their various implications, it is possible to deduce the desirable circumstances for each scheduling action and hence select the most appropriate scheduling action to apply in a given situation. The results are summarized in Figure 3-1 below.

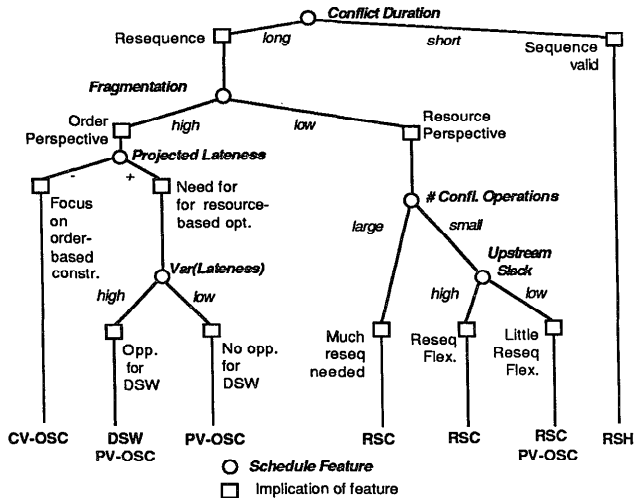

Figure 3-1: Decision tree for selecting actions

As shown in Figure 3-1, when the conflict duration is short (implying that the sequencing decisions of the current schedule are still valid), RSH is postulated as the most appropriate action to take. This is because RSH resolves conflicts while maintaining the stability of the sequence in an efficient manner. There is a need for some reoptimization when the conflict duration is long. When the conflict centers on a resource with low or no fragmentation (i.e. a bottleneck resource), a resource perspective is needed to ensure that critical resource-based constraints and goals are optimized by the reaction. Conversely, a highly fragmented resource schedule provides an opportunity for order-based optimization. To select a particular scheduling action within each perspective requires further analysis of the current control state.

If a resource-based reaction is appropriate, there are two possible actions: RSC and PV-OSC (with the scope of the action limited to just the conflicting operations). If either the number of conflicting operations is high or there is upstream

slack that can be exploited for resequencing purposes, then RSC is the most efficient and effective reactive action that can be taken. This follows from its strength in optimizing the utilization of a particular resource. However, if only one or two conflicting operations are present and there is little upstream slack, then PV-OSC may be sufficient. In this case the resequencing problem is constrained to one of simply repositioning the conflicting operation(s) within the focal point resource's schedule.

Under the order-based perspective, we may choose between three actions: CV-OSC, PV-OSC and DSW. If there is plenty of slack in meeting the due date of an order involved in a conflict, the CV-OSC is preferable as it minimizes disruption to the existing schedule without threat of the order being tardy. However, if the conflict compromises the quality of the schedule relative to order tardiness, a more aggressive approach to order scheduling is needed. When there is a high variance in the projected lateness of jobs, an opportunity may exist to swap demands with DSW. Note however, that DSW acts more to improve the nature of the current conflict, and as a rule is not used more than once per externally generated conflict. PV-OSC provides a consistent, more aggressive approach to order scheduling.

One additional consideration in selecting actions which is not reflected in Figure 3-1 is the scope of the reaction. In the case of resource-based reactions, the scope is naturally the focal point resource. However, in the case of order-based reactions, the scope should depend on extent of resource contention further downstream. Specifically, if downstream slack is high (indicating the presence of downstream bottleneck resources), then the scope of an order-based scheduling action is limited to the portion of the order's production plan that precedes the downstream bottleneck operation. This provides the opportunity to take full advantage of the strengths of resource-based scheduling actions.

3.3. Experiments

To test the proposed model for selecting scheduling actions, a series of experiments involving a specific set of reactive problems has been designed. Each reactive problem is defined by generating a "starting" schedule, establishing a current "state of execution" at some point within the schedule and then introducing either a machine breakdown or an operation processing failure (implying extra repair operations). The problems have been specified so as to ensure that all combinations of the schedule features are enumerated. Each experiment then consists of applying alternative scheduling actions to each problem. To empirically verify that the proposed model is correct, it is necessary to show that its prescribed action performs as well or better than actions not prescribed by the model. An added complication occurs because certain actions chosen may not leave the schedule conflict-free. It is not possible to evaluate the quality of a schedule with conflicts, so that for these situations, the performance evaluation applies to a series of reactive scheduling actions. For these cases, it is only possible to test that the successive actions as prescribed by the model give equal or better performance than deviations from the model.

Table 3-3 summarizes the experimental results that have been obtained to date. It shows, for each reaction cycle, both the action(s) that performed best according to the criteria stated earlier and the action prescribed by the model. As can be seen, the actions prescribed by the model in each experiment were among the best.

4. Discussion

In this paper, we have advocated an opportunistic methodology for reactive scheduling based on focused

Schedule Features*							Actions	
Exp. #	Cycle	Conflict Duration	Fragmen-tation	Upstream Slack	Downstream Slack	Lateness	Best Actions	Prescribed Action
1,3	1	short	high	no	no	negative	RSH, CV-OSC	RSH
2	1	long	moderate	no	no	negative	RSH, CV-OSC	CV-OSC
4	1, 1/2	short	high	yes	no	negative	RSH, RSC/RSH	RSH
5-8	1	long	low	yes	yes	positive	RSC	RSC
	2	short	high	yes	no	positive	RSH	RSH
9	1	long	low	yes	yes	negative	RSC	RSC
	2	long	high	no	yes	negative	PV-OSC, CV-OSC	CV-OSC
	3	long	low	yes	no	positive	RSC	RSC
	4	long	high	no	no	positive	PV-OSC, CV-OSC	PV-OSC
10	Verified that DSW is able to take advantage of wide lateness variance.							
11	Verified that RSC is better than PV-OSC when *upstream slack* is present.							

* All schedules have wide lateness variance.

Table 3-1: Experimental Results

analysis of the conflicts introduced into an existing schedule. The methodology presumes the availability of a set of alternative scheduling actions, each of which operates with respect to a particular local perspective of the problem and offers selective advantages for conflict resolution. We have focused specifically on the meta-level control problem, considering the issue of conflict analysis and presenting a model for selecting among potential actions. Before closing, we briefly discuss those aspects of the overall reactive problem that have not been addressed.

Our model of conflict analysis and reaction selection emphasizes schedule quality. Efficiency concerns are addressed only to the extent that disruption to the starting schedule is minimized by revising only those scheduling decisions that are found to be invalid, and minimizing disruption is correlated to reactive efficiency. However, the pragmatics of reacting within a specified time frame have clearly not been addressed. In this regard, two points can be made:

- One characteristic of the model is that revision of the schedule proceeds forward in time. Once the first scheduling action has been completed, the immediate conflict has been solved and revision is now centered on responding to the downstream consequences of this action. Thus, the immediately needed scheduling decisions are now valid. If necessary, the system could suspend work on the residual problems and focus on other more pressing matters. This requires a framework for prioritizing and managing pending conflicts. Here, recent work in reactive planning architectures [Georgeff 87, Firby 87] is directly relevant.
- The efficiency of individual scheduling actions can be influenced by a number of factors, including level of abstraction of the problem/schedule, search parameters, and the use of problem-specific heuristics.

Acknowledgements

We gratefully acknowledge the long hours spent by Dirk Matthys and Jean-Yves Potvin in running the experiments and their help in analyzing the results. Thanks also to Chris Young for his work in developing the OPIS testing environment.

References

[Agre&Chapman 87]
Agre, P.E. and D. Chapman.
Pengi: An Implementation of a Theory of Activity.
In *Proc. AAAI-87*, pages 268-272. Seattle, WA, July, 1987.

[Bean&Birge 85] Bean, J.C., and J.R. Birge.
Match-Up Real-Time Scheduling.
Technical Report 85-22, Univ. of Michigan Dept. of Industrial and Operations Engineering, June, 1985.

[Firby 87] Firby R.J.
An Investigation into Reactive Planning in Complex Domains.
In *Proc. AAAI-87*, pages 202-206. Seattle, WA, July, 1987.

[Fox&Smith 84] Fox, M.S., and S.F. Smith.
ISIS: A Knowledge-Based System for Factory Scheduling.
Expert Systems 1(1):25-49, July, 1984.

[Georgeff 87] Georgeff M.P.
An Embedded Real-Time Reasoning System.
In *Proc. 2nd Annual NASA Artificial Intelligence Research Forum*, pages 286-329. Palo Alto, CA, 1987.

[Johnson 74] Johnson, L.A. and D.C. Montgomery.
Operations Research in Production Planning, Scheduling and Inventroy Control.
John Wiley, 1974.

[Lansky&Fogelsong 87]
Lansky, A.L. and D.S. Fogelsong.
Localized Representation and Planning Methods for Parallel Domains.
In *Proc. AAAI-87*, pages 240-245. Seattle, WA, July, 1987.

[LePape&Smith 87]
LePape, C. and S.F. Smith.
Management of Temporal Constraints for Factory Scheduling.
In C. Rolland, M. Leonard, and F. Bodart (editors), *Proc. IFIP TC 8/WG 8.1 Working Conf. on Temporal Aspects in Information Systems (TAIS 87)*, pages 165-176. Elsevier Science Publishers, May, 1987.

[Ow&Smith 88] Ow, P.S. and S.F. Smith.
Viewing Scheduling as an Opportunistic Problem Solving Process.
In R.G. Jeroslow (editor), *Annals of Operations Research: Approaches to Intelligent Decision Support*, pages 85-108. Baltzer Scientific Publishing Co., 1988.

[Smith 87] Smith, S.F.
A Constraint-Based Framework for Reactive Management of Factory Schedules.
In M. Oliff (editor), *Proc. 1st Inter. Conf. on Expert Systems and the Leading Edge in Production Management*, pages 349-366. Charleston, SC, May, 1987.

[Smith&Ow 85] Smith, S.F. and P.S. Ow.
The Use of Multiple Problem Decompositions in Time-Constrained Planning Tasks.
In *Proc. IJCAI-85*, pages 1013-1015. Los Angeles, CA, August, 1985.