CHAPTER 1

# Directing an opportunist scheduler: an empirical investigation on reactive scenarios

*G. Hasle[a] and S.F. Smith[b]*

[a]*Section of Knowledge-based Systems, SINTEF Informatics, P.O. Box 124, 0314 Oslo, Norway*

[b] *Center for Integrated Manufacturing Systems, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213–3890, USA*

**Abstract**

Opportunistic scheduling offers a uniform perspective on predictive and reactive scheduling as iterative problem solving processes. In the context of reactive scheduling, it constitutes a knowledge-directed alternative to more search intensive iterative approaches. By adopting a scheduling process that opportunistically focuses attention on the most critical subproblem and carefully selecting the focal point of the next problem solving effort, one can significantly constrain search while continuing to give attention to important scheduling objectives. Thus, one can maintain high-quality solutions in the face of changing constraints under stringent response time constraints. Control heuristics implemented in an architecture for opportunistic control determines the identification, analysis, and prioritization of subproblems, as well as the formulation of problem solving tasks. The nature of the control architecture determines the the span of control heuristics that may be accommodated. In addition to the repertoire and nature of methods for subproblem resolution, the nature of control heuristics plays a critical role in the performance of opportunistic scheduling systems. This paper describes a novel control architecture which represents a generalization of earlier architectures for opportunistic scheduling. It accommodates what we have denoted as *focal point-opportunistic scheduling strategies.* New control heuristics that draw upon the extended expressiveness of the novel control architecture are presented, as well as results from a comparative, empirical investigation of these heuristics based on reactive scenarios for a rich factory model.

Keyword Codes: I.2.1; I.2.4; I.2.8
Keywords: Applications and Expert Systems; Knowledge Representation Formalisms and Methods; Problem Solving, Control Methods and Search

R. Kerr et al. (eds.), *Artificial Intelligence in Reactive Scheduling*

## 1. INTRODUCTION

The development of techniques for incremental, iterative schedule repair and improvement has received increasing attention in recent years. Such techniques can be broadly characterized by their reliance on incomplete, local search procedures; search is focused within particular solution "neighborhoods", generally (but not necessarily) through manipulation of an existing seed schedule. One reason for this trend stems from the basic complexity of scheduling problems. In practical settings, schedules must typically satisfy a diverse set of domain constraints while attending to a range of (typically conflicting) performance objectives. Search is fundamental to the achievement of acceptable solutions; but underlying combinatorics typically discourage use of systematic global backtracking-search procedures. A second reason for increased emphasis on schedule repair and improvement techniques reflects the characteristics of practical scheduling problems. In most application environments, the development of schedules is an iterative and ongoing process; initial solutions are developed, deficiencies are noted, requirements and constraints are renegotiated, solutions are revised, etc. Throughout this process, the current schedule provides an important decision-making reference and an ability to preserve solution continuity across iterations has considerable pragmatic value. And as execution unfolds, domain unpredictability continually forces changes to prescribed plans; in this context, continuity constraints are equally important and response time requirements become more critical.

One central issue in the design of schedule repair and improvement procedures is how to effectively focus the search process, and one can distinguish between alternative approaches on the basis of the knowledge and bias that is incorporated to direct the search. Toward one end of the spectrum, more search intensive approaches have been investigated, which place relatively little computational effort into explicitly directing the search, and rely instead on exploring and evaluating a sizable set of solutions. Movement in the space occurs through (typically) non-deterministic application of simple revision operators (e.g., [9, 17]) and the principal search bias is provided in the form of an evaluation function. In some cases (e.g., [1]), knowledge of previously encountered solutions is also incorporated to further concentrate the search toward profitable regions. Alternatively, other research has emphasized more deliberate, knowledge-directed approaches, which expend more computational effort reasoning about and structuring the revision/improvement process, and generate far fewer alternative solutions. So-called *opportunistic* scheduling techniques (e.g., [11, 5, 2, 3, 13]), which reason about the structure and implications of current solution constraints to determine regions of the schedule that require change, revision objectives to emphasize and appropriate revision procedures (or operators) to apply, fall into this category. There are tradeoffs associated with either design perspective. Knowledge-directed approaches have demonstrated the possibilities for efficient, localized schedule revision in response to constraint changes with continuing attention to domain performance objectives (e.g., [12, 15]). But, these approaches can also be susceptible to overly restrictive search bias (a function of the system's control knowledge and heuristics), which can result in missed revision/improvement opportunities. Broader search-based approaches, alternatively, offer a more flexible basis for focusing the search (a function of the system's evaluation function) and less susceptibility to the limits of a priori conceived revision knowledge and strategies, but solution quality and computational cost is less predictable (particularly in the presence of multiple performance objectives and preferences) and generally depends significantly on the quality of seed solutions.

In this paper, we adopt a knowledge-directed perspective on the schedule revision and improvement problem, and describe ongoing work aimed at the development of control architectures and knowledge for intelligent revision of schedules in response to changing circumstances. We take, as our starting point, the opportunistic reasoning framework originally developed within the OPIS manfuacturing scheduling system [16], and the search control strategies originally evaluated in [12, 15] (hereafter referred to as OPIS/AAAI-88 whenever appropriate). We focus on the constituent process of analyzing and selecting *focal points* for subproblem formulation at each iteration of the schedule revision process, and identify limiting aspects of original architectural commitments. A revised control architecture is proposed, control heuristics that draw upon this architecture's extended expressiveness are summarized, and initial results of a comparative experimental analysis of performance advantages in various reactive scenarios is presented. To provide a context for describing this work, we begin by summarizing relevant aspects of the OPIS scheduling architecture.

## 2. THE OPIS SCHEDULING ARCHITECTURE

In OPIS, reactive (and predictive) scheduling proceeds opportunistically as an iteration of problem state analysis, i.e., the identification and characterization of *control events* (i.e., bottlenecks, inconsistencies, opportunities, and incompleteness) by *Analysis Knowledge Sources* (AKSs), and the subsequent formulation and execution of tasks based on a repertoire of *Scheduling Knowledge Sources* (SKSs). Analysis of candidate subproblems in the form of control events is typically based on constraint metrics. Analysis results are used to select the subproblem which seems most important to pursue next and to formulate a task to resolve it. A blackboard-oriented control architecture is used to coordinate system activity, which organizes subproblem formulation knowledge in terms of *control heuristics* for

- problem state analysis
- subproblem prioritization
- task formulation

The blackboard contains a representation of the current schedule, which is accessed by the AKSs and SKSs. Scheduling commitments made by SKSs are propagated to related decisions by a schedule maintenance system, and control events (e.g., constraint conflicts) are identified and posted to a control manager. A particular configuration of AKSs, SKSs and control heuristics is denoted a *scheduling strategy.*

Among the SKSs defined in OPIS are:

- an order scheduler, which is capable of making or changing scheduling decisions associated with a given manufacturing order and is biased toward lead time minimization,
- a resource scheduler, which is capable of making or changing scheduling decisions associated with a resource (or set of substitutible resources) and emphasizes efficient resource utilization, and
- a temporal shifter, which can be used to remove inconsistencies by shifting decisions forward or backward in time.

AKS defined within OPIS include:

- capacity analysis, which estimates expected resource contention levels,
- an inconsistency analyzer, which computes measures of temporal and resource flexibility relative to sets of decisions currently in conflict, and
- an aggregation analyzer, which recognizes relationships among the decisions associated with distinct control events that may suggest their simultaneous consideration.

Details of the scheduling architecture and its components may be found in [16].

## 3. FOCAL POINTS IN OPPORTUNISTIC SCHEDULING

In opportunistic reasoning, an important issue in task formulation is the determination of the appropriate *focal point* (i.e., what set of decisions to focus on). Generally speaking, knowledge sources (both AKSs and SKSs) have one or more degrees of freedom for delimiting their problem solving effort. These degrees of freedom are related with the structural characteristics of the problem domain as well as the particular problem solving perspective of the knowledge source.

For the knowledge sources described above there are three basic focal point dimensions. Let us describe these dimensions and give examples of the effects of focusing in these dimensions.

- temporal extent
  The analysis of a control event may be restricted to a particular temporal interval centered on the event and thus give a more or less local characterization. A temporal shifter task may be restricted to shift operations within a certain temporal interval.
- resource hierarchy level
  In a hierarchical resource model, the value of constraint metrics (e.g., resource contention) may be highly dependent on the level at which they are evaluated, as the scope of resources is widened as one moves up the resource hierarchy (e.g., from individual machines to substitutible machine groups). Given a temporal extent, a resource scheduler which searches for optimal allocations on a focal point resource will be more constrained in its search when one moves down the hierarchy.

- the upstream/downstream dimension
  An inconsistency event resulting from the violation of a precedence constraint has (at least) two alternative focal points (upstream and downstream) related with precedence links, both for analysis and task formulation.

## 4. AN ARCHITECTURE FOR FP-OPPORTUNISTIC SCHEDULING

Although the OPIS KSs contain some flexibility to focus attention in the three focusing dimensions, the OPIS control architecture may only to a limited degree accommodate control heuristics that take advantage of this flexibility. By generalizing the OPIS control architecture in the way described below we have increased the flexibility to accommodate scheduling strategies that dynamically select the focal point of reaction (and thus also the the granularity of control decisions) on the basis of a more comprehensive problem state analysis at alternative focal points. Such scheduling strategies may be regarded as a generalization of earlier opportunistic scheduling strategies. We shall denote this approach *focal point-opportunistic scheduling* or *FP-opportunistic scheduling.*

One perceived problem with the OPIS control architecture is *early commitment to focal point event.* The analysis of alternative subproblems is performed at a single focal point, using a "static"[1] focal point selection heuristic. Prioritization and task formulation (including focal point selection) may consequently suffer from uninformed decisions, which again can result in missed opportunities for more effective schedule revision. In response to this problem, the control architecture is reorganized to contain the following steps:

- posting of control events
- control event aggregation
- event analysis
- focal point selection
- event prioritization
- task formulation
- task execution and constraint propagation

Control events are aggregated to treat related control events as a single event and thus achieve a more adequate and less nervous reaction. Event analysis is performed in order to base task formulation on relevant constraint metrics, either evaluated directly from the current state, or generated through probabilistic look-ahead analysis.

The analysis of events at multiple focal points provides more information to the focal point selection and event prioritization steps. In opportunistic scheduling, inconsistencies may arise both during predictive scheduling (due to the opportunistic focus of attention)

[1] I.e., one which is not based on an analysis of the current problem solving state.

and reactive scheduling (also due to unexpected external events). Hence, the search process may jump between the feasible and non-feasible regions of the search space[2].

Resource capacity conflicts (RCs) and temporal precedence constraint violations (PVs) are examples of control events of the inconsistency type. In a hierarchical factory model, an RC detected at the lowest (machine) level may be analyzed at every node in the branch of the resource hierarchy which leads to this machine. The constraint metric values, and the gradient of these values over the resource hierarchy constitute important information to determine focal point in the resource level dimension. PVs have an upstream and downstream commitment. The analysis of PVs both at the upstream and downstream focal points (and at every relevant level in the resource hierarchy) may again result in a better selection of focal point for PVs. Generally, the event prioritization and task formulation steps in opportunistic scheduling will benefit from a more informed selection of focal point. The benefits of more comprehensive problem state analysis must be compared with the extra overhead incurred.

## 5. AN FP-OPPORTUNISTIC CONTROL STRATEGY

We have designed *FOCS-0,... ,FOCS-3*, a suite of four FP-opportunistic control strategies that draw upon the enhanced capabilities of the generalized control architecture. They are designed for use with the OPIS SKSs. The following section describes a empirical investigation of *FOCS-0* on a rich factory model. Our benchmark is the OPIS architecture instantiated with *AAAI-88*, a particular control strategy described in [12]. *FOCS-0* is a basic strategy which includes:

- a new event aggregation heuristic
  In the OPIS/AAAI-88 configuration events were deemed related on the basis of commonality of resource focal point only. No discrimination was made on the basis of temporal separation of events. Hence, events that are located at widely separated points in time in the schedule were unconditionally aggregated with a possible effect of a drastic and time-consuming reaction. The *FOCS-0* heuristic determines aggregation of a set of potentially related events also by considering temporal separation and contention metrics on the resource in question.

- a revised definition of metrics for conflict events
  Several metrics utilized to characterize constraint conflicts in OPIS are parameterized to limit attention to a particular subhorizon of the overall schedule. In the OPIS/AAAI-88 configuration, these temporal scope settings were restricted to the specific temporal interval of the conflict (giving no attention to the local "neighborhood" in the schedule surrounding the conflict). For example, in estimating resource utilization in the event of a machine breakdown, no consideration was given to projected usage of the resource(s) after the projected end time of the breakdown. In *FOCS-0*, the temporal scope of the calculation of contention has been extended, and the calculation is based on averages rather than peak values.

[2]One may argue that the freedom to enter non-feasible regions may give benefits over the more constrained search processes generated by standard CSP techniques.

- a new focal point selection heuristic
  In contrast with the "static" OPIS/AAAI-88 focal point selection heuristic, *FOCS-0* determines a more informed focal point determination on the basis of event analyses on a set of alternative focal points. Focal point selection in the resource hierarchy dimension is performed opportunistically according to the gradient of resource contention over resource hierarchy level. The motivation is to take care of the following phenomenon: A conflict which is deemed serious at the lowest (machine) level and consequently calls for a major, disruptive and time-consuming reaction, may be assessed as trivial (and hence be solved in a better way using a minor, less disruptive and faster reaction) at higher resource hierarchy levels.

- a new event prioritization heuristic
  In OPIS/AAAI-88, event prioritization is largely based on a static prioritization of event types. For equally typed events, there are tie-breakers that draw upon event metrics at the statically selected focal point. In *FOCS-0*, event prioritization is performed by comparing several event metrics at the dynamically selected focal point.

As a consequence of these changes in control heuristics, a few minor changes in the heuristic for selection of knowledge source were needed.

The remaining strategies are variants of *FOCS-0* where precedence conflicts are analyzed and may be aggregated at both alternative focal points. The remainder of this paper will be focused on *FOCS-0*.

## 6. EMPIRICAL INVESTIGATION

An empirical and comparative investigation of *FOCS-0* has been performed. The revised architecture instantited with *FOCS-0* was compared with AAAI-88. To this end the *WS-model*, a rich model of a factory which has been investigated in earlier experiments[3] was selected. The WS-model describes a factory of the job-shop type with alternative resources and sequence-dependent setup times. A series of 22 predictive scenarios (typically comprising some 120 orders and 500 operations) describing a large variety of shop conditions and order priorities has been generated and used as a benchmark for experiments comparing an early version of OPIS, ISIS, and the COVERT dispatch heuristic [14]. The predictive scenarios were used as a preamble to experiments with reactive scenarios. For the subset of experiments where conflicts are generated the reactive scheduling strategies come into play and may thus be partially evaluated. The performance of OPIS/AAAI-88 and *FOCS-0* was compared w.r.t. the weighted tardiness (TCO), work-in-process (WIP), and response time (CPU) criteria. Differences were on the average small. 3 experiments created no reaction, 10 experiments created identical scheduling processes, on 3 counts *FOCS-0* generated dominating schedules faster, in one case *FOCS-0* generated a dominating schedule slower. In the remaining 6 experiments tradeoffs between the evaluation criteria were observed.

[3]The WS-model is a rich model of the Westinghouse turbine plant in Winston-Salem, NC. It is described in [10], [4].

A single schedule generated from predictive scenarios was selected as the starting point for reactive scenarios[4]. For this schedule 20 machine breakdown scenarios were generated randomly on four different machines:

- a machine for the most upstream operation
- a machine in a typical primary bottleneck area
- a machine in a typical secondary bottleneck area
- a machine for the most downstream operation

Breakdown times were drawn from a uniform distribution $U(t_1, t_2)$ covering the makespan on the machine in question, i.e.,

$$t_1 = \min_{o \in O_M} \{st(o)\}$$

and

$$t_2 = \max_{o \in O_M} \{et(o)\}$$

$st(o)$ and $et(o)$ is the scheduled start and end time, respectively, for operation $o$. $O_M$ is the set of operations on resource $M$. Breakdown durations were drawn from the uniform distribution:

$$U(0.5\,\overline{dur}(O_M), 4.5\,\overline{dur}(O_M))$$

where $\overline{dur}(O_M)$ is the average duration of the operations on M. Comparative experiments were run and statistics gathered on schedule quality (TCO, WIP), response performance (CPU), and schedule disruption (DIS). Table 1 summarizes the results from the reactive experiments. For each resource WIP, TCO, DIS, and CPU averages over the 20 breakdown scenarios are shown for the *AAAI-88* and *FOCS-0* strategies. Total number of operations moved is used as the disruption measure[5]. Note the large average quality (31% WIP,

| **AAAI-88 (old) vs. FOCS-0 (new)** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *Res.* | $WIP_{old}$ | $WIP_{new}$ | $TCO_{old}$ | $TCO_{new}$ | $DIS_{old}$ | $DIS_{new}$ | $CPU_{old}$ | $CPU_{new}$ |
| *Upstr.* | 411.03 | 283.35 | 52.24 | 10.62 | 96.35 | 34.85 | 334.05 | 471.20 |
| *Prim.BN* | 288.93 | 283.42 | 12.50 | 11.32 | 255.40 | 238.15 | 2087.30 | 1994.50 |
| *Sec.BN* | 283.26 | 296.65 | 10.68 | 13.07 | 39.20 | 54.00 | 361.05 | 526.03 |
| *Dnstr.* | 275.71 | 275.09 | 10.18 | 10.23 | 11.70 | 6.85 | 166.30 | 38.45 |

Table 1: Results from Reactive Experiments

80% TCO) and disruption (64%) improvement for breakdowns on the upstream resource. These improvements came with the price of a 40% CPU increase. When scrutinizing the results of individual experiments it was discovered that there are large variations. In a few cases drastically better schedules were generated much faster and with substantially less

[4]The selected experiment showed quality tradeoff and speed degradation for *FOCS-0* in the predictive scheduling experiments. The OPIS/AAAI-88 variant was incidentally selected.

[5]The results on more sophisticated disruption criteria will be available later.

disruption. It was verified[6] that FP-opportunism was the direct cause of improvement in these cases. Further analysis (including classification of the breakdown scenarios on the basis of constraint metric values) will be needed to provide more general conclusions.

For the primary bottleneck experiments there are uniform, but small improvements (1.9% WIP, 9.5% TCO, 6.8% DIS, 4.4% CPU). This observation may be explained by the fact that the original schedule is tight, not only for the broken machine, but for the whole workarea of interchangeable machines. FP-opportunism in the resource level dimension has little to offer in this situation[7].

For the secondary bottleneck, the average results are significantly worse (4.7% WIP, 22.4% TCO, 37.8% DIS, 45.8% CPU) for *FOCS-0*. Individual results show substantial variations, and there are 3 cases where *FOCS-0* show drastic improvements on all criteria. Again, more analysis is needed.

The downstream resource experiments on the average show negligible schedule quality differences, but with large improvements of disruption and response performance (41.5% and 77%, respectively). These results are uniform with a few exceptions of cases where the two control strategies generate identical scheduling behaviour.

The experiment series revealed a few unexpected but interesting results. In isolated cases, the repair of a machine breakdown event produces a schedule which dominates the original, indicating a potential for scheduling strategy improvement. As an additional remedy in opportunistic scheduling we propose the addition of *optimizing knowledge sources (OKSs)*. OKSs could typically operate in an anytime fashion [17] by performing neighborhood search on complete schedules.

## 7. CONCLUSION

We have investigated the potential of FP-opportunistic scheduling strategies by conducting comparative empirical investigations with machine breakdown scenarios on a rich factory model. Preliminary results show a potential for large improvements over earlier opportunistic scheduling strategies by taking advantage of the *resource level* degree of freedom for focal point selection. Further refinement is needed, as well as the development and experimental investigation of FP-opportunistic scheduling strategies that draw upon other degrees of freedom for focal point selection.

In the introduction we mentioned the current dichotomy between knowledge-directed and search intensive approaches to reactive scheduling as well as the potential strengths and weaknesses of each. In general, we believe that the predominantly knowledge-directed, opportunistic scheduling approach discussed in this paper (which is targeted at creating and maintaining high quality schedules under real time response requirements essentially without backtracking) could be enhanced by including knowledge sources that perform anytime optimization on the basis of more search intensive methods. Some empirical results reported in the previous section further supports this belief. Techniques from the area currently known as *modern heuristics* (e.g., tabu search [8, 1]) seem natural candidates for the underlying methods. In the context of an opportunistic scheduling

[6] Through a rather cumbersome inspection of traces.

[7] Note that we have restricted ourselves to comparison of scheduling strategies that only differ in terms of control strategy.

framework, such schedule improvers could be implemented in a natural way as separate knowledge sources that would typically be invoked as a response to internally generated *opportunity events* as well as externally generated desires to improve schedule quality components.

# References


[1] Barnes J.W., Laguna M., Glover F.: An Overview of Tabu Search Approaches to Production Scheduling Problems. To appear in *Intelligent Scheduling Systems* (eds.) Brown, and Scherer, Kluwer Pub., 1994.

[2] Bensana E., Bel G., Dubois D.: OPAL: A Multi-knowledge Based System for Industrial Job Shop Scheduling. *International Journal of Production Research* 26(5), 1988, pp 795-815.

[3] Berry P.M.: Satisfying Conflicting Objectives in Factory Scheduling. *Proc. The Sixth Conference on Artificial Intelligence Applications*, Santa Barbara, CA, March 1990, pp 101-106.

[4] Chiang W.Y., Fox M.S., Ow P.S.: Factory Model and Test Data Descriptions: OPIS Experiments. Tech. Report CMU-RI-TR-90-05, Carnegie Mellon University, Pittsburgh PA, 1990.

[5] Collinot A., Le Pape C., Pinoteau G.: SONIA: A Knowledge-Based Scheduling System. *Artificial Intelligence in Engineering*, Vol 3, No 2, 1988, pp 86-94.

[6] Engelmore R.S., Morgan A.J.: *Blackboard Systems.* Addison-Wesley Publishers Ltd., ISBN 0-201-17431-6.

[7] Fox M.S., Smith S.F.: ISIS - A Knowledge-Based System for Factory Scheduling. *Expert Systems*, Vol. 1, No. 1, 1984.

[8] Glover F.: Artificial Intelligence, Heuristic Frameworks and Tabu Search. *Managerial and Decision Economics*, Vol. 11, 1990, pp 365-375.

[9] Minton S., Johnston M.D., Phillips A.B., Laird P.: Solving Large-Scale Constraint Satisfaction and Scheduling Problems Using a Heuristic Repair Method. *Proceedings 8th National Conference on Artificial Intelligence (AAAI-90)*, Boston, MA, July, 1990, pp 17-24.

[10] Ow P.S.: Experiments in Knowledge-Based Scheduling. Tech. Report, Carnegie Mellon University, Pittsburgh PA, April 1986.

[11] Ow P.S., Smith S.F.: Viewing Scheduling as an Opportunistic Problem Solving Process. *Annals of Operations Research*, Vol. 12, (ed) R. Jareslow, Baltzer Scientific Publishing, 1988, pp 85-108.

[12] Ow P.S., Smith S.F., Thiriez A.: Reactive Plan Revision. *Proc. AAAI-88*, St. Paul, Minn., USA, 1988, pp 77-82.

[13] Sadeh-Koniecpol N.: Look-Ahead Techniques for Micro-Opportunistic Job Shop Scheduling. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, 1991.

[14] Smith S.F., Ow P.S., LePape C., McLaren B., Muscettola N.: Integrating Multiple Scheduling Perspectives to Generate Detailed Production Plans. *Proc. 1986 SME Conference on AI in Manufacturing*, Long Beach, CA.

[15] Smith S.F.: An Integrated Framework for Generating and Revising Factory Schedulings. *Journal of the Operational Research Society*, Vol. 41, No. 6, (June 1990), pp. 539-552.

[16] Smith S.F.: OPIS: An Architecture and Methodology for Reactive Scheduling. in *Intelligent Scheduling*, (eds) Zweben M. and Fox M.S., Morgan Kaufmann Pub., Palo Alto, 1994.

[17] Zweben M., Deale M., Gargan R.: Anytime Rescheduling. *Proceedings 1990 DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, (ed) Sycara K.P., Morgan Kaufmann Pub., San Diego, CA, November, 1990, 251-259.