Integrating Planning and Scheduling: Towards Effective Coordination in Complex, Resource-Constrained Domains*†

Stephen F. Smith
The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

November 10, 1994

Abstract

In this note, we summarize current research at CMU aimed at extending constraint-based scheduling frameworks and heuristics to enable effective integration of resource allocation and plan synthesis processes. Similar to prior work in opportunistic scheduling, our approach assumes the use of dynamic analysis of problem space structure as a basis for heuristic focusing of problem solving search. This methodology, however, is grounded in representational assumptions more akin to those adopted in recent temporal planning research, and in a problem solving framework which similarly emphasizes constraint posting in an explicitly maintained solution constraint network. We summarize recent experimental results that indicate that such problem formulation assumptions can in fact lead to better heuristic solutions than have been obtained with more-classical assignment problem formulations on benchmark problems previously studied within both the Artificial Intelligence and Operations Research communities. We conclude with a brief discussion of some important open research questions in this area.

1 Introduction

Many important practical problems require efficient allocation of resources to competing goal activities over time in the presence of complex state-dependent constraints. Synchronizing the on-board activities of a space mission, coordinating the movement of personnel and supplies to support disaster relief efforts, and managing the flow of materials through an automated manufacturing facility are all examples of this type of problem. Such problems are typically categorized as scheduling problems, where resources must be allocated so as to optimize overall performance objectives (e.g.,

†This research has been sponsored in part by the National Aeronautics and Space Administration, under contract NCC 2-531, by the Advanced Research Projects Agency under contract F30602-90-C-0119 and the CMU Robotics Institute.

*Invited, keynote talk at the 1993 Italian Planning Workshop, Rome, Italy, September, 1993.

maximizing scientific return of space missions, initiating relief efforts as soon as possible, maximizing product throughput). At the same time, since the executability of a given goal activity in such problems also depends on conditions of the predicted world state other than resource availability (e.g., spacecraft vibration level, the locations of transport or material handling vehicles), solution feasibility can only be guaranteed by dynamically generating and synchronizing the auxiliary activities necessary to bring about and preserve enabling state conditions. In short, effective solutions to these problems must integrate resource allocation and plan synthesis capabilities.

2 Scheduling and Planning Approaches

Existing scheduling and planning technologies provide inadequate solutions to this class of problems. Scheduling research has generally focused on the question of *"When"* activities should be executed: Given a set of activities to be scheduled and their associated temporal constraints (e.g., the production plans for a set of jobs to be scheduled), how should required resources be allocated over time so as to satisfy process and resource capacity constraints, and simultaneously optimize some objective criterion (e.g., minimizing tardiness, minimizing overall schedule duration, etc.). This perspective leads naturally to formulation of scheduling problems as assignment problems (i.e., a problem of assigning resources and start times to goal activities)[Bak74]. Unfortunately, such formulations do not acknowledge the general presence of state-dependent constraints nor the need to generate "state changing" activities in order to satisfy them. Recent work in constraint-based scheduling e.g., [SOMM90, ZDG90]) has extended classical formulations to support treatment of special classes of state-dependent constraints other than resource availability (e.g., sequence dependent machine setups, enforcement of enabling conditions other than resource availability). However, this work has relied on representations that support restricted, localized forms of disjunction in goal activity specification (e.g., alternative production processes/technologies for a given manufacturing step), which can be introduced while retaining an assignment formulation of the problem. In many domains, these representations have proved sufficient. However, in many other domains, the complexity (e.g., non-locality) of state-dependent constraints makes such approaches impractical.

AI planning research, alternatively, has focused principally on the question of "*What*" activities to execute, developing mechanisms for configuring activity networks to achieve goals from more basic descriptions of how various domain actions affect the world. In these representational frameworks, there are no a priori restrictions on the complexity of state descriptions, and more recent work in temporal planning [AK83, DM87] has removed the impoverished assumptions of classical planners (e.g., [FHN72]) concerning representation of actions in time. However, with few exceptions (e.g., [Lan88]) this work has ignored the issue of problem/domain structure (e.g., representing state instead as an unstructured set of predicates). Use of problem structure has been central in scheduling research and a key to developing effective optimization heuristics in large-scale domains. Recent work in opportunistic scheduling [SOMM90, Sad91] provides a good example. By assuming that the dynamic state of the world is structured as a set of resource availability "state variables", estimation of resource contention can be exploited to profitably direct the solution process. While there are some examples of AI planners that have successfully tackled realistic applications (e.g., Wilkins88), there are just as many examples where lack of emphasis on problem structure has led to insurmountable search difficulties (e.g., the Voyager mission planning experiences with DEVISER [Ver83]).

3 Integrated Planning and Scheduling Architectures

Our work toward synthesizing the respective strengths of planning and scheduling frameworks has led to the development of HSTS: an integrated planning and scheduling architecture [MSCD92, Mus93b]. There are three distinguishing characteristics of the HSTS framework:

1. A representational framework that decomposes the state of the world into a finite set of "state variables" which vary over time, and describes domain dynamics (e.g., activity pre and post conditions) in terms of temporally "compatible" state variable value configurations. The modeling framework thus integrates the problem/domain structure inherent in scheduling representations with the expressiveness of modern temporal planning frameworks. This enables the specification of scheduling algorithms that exploit problem decomposability and provides the necessary structure for optimizing resource utilization.

2. A flexible representation of solutions (i.e., possible executions) as an explicit temporal constraint graph, extending the temporal data base concept of [DM87] to incorporate state variable structure. Within this solution model, the occurrence of events (e.g., activity start times) can be allowed to "float" within the temporal constraints imposed by the problem and the problem solving process. This avoids the problems of over-commitment inherent in "fixed times" scheduling frameworks and contributes directly to increased scheduling efficiency (see Section 4 below).

3. A uniform view of planning and scheduling processes as an iterative constraint posting process. Flexibility is provided to accommodate a range of problem solving strategies (e.g. forward simulation, back chaining, etc.) and to support dynamic interleaving of goal sequencing and goal expansion actions. This allows the incorporation of algorithms that opportunistically exploit problem structure to consistently direct problem solving toward the most critical tradeoffs that need to be made.

The HSTS problem solving architecture was originally developed and applied in the context of the problem of constructing short-term observation schedules for the Hubble Space Telescope (HST), motivated by the limitations of the current solution and, more generally, by the insufficiency of classical planning and scheduling approaches in this problem context. In the HST domain, several results with the HSTS problem solving architecture have been demonstrated. The leverage provided by HSTS's emphasis on decomposable domain descriptions was demonstrated through experiments with a sequence of domain models that increasingly captured more and more of the telescope's operational constraints. The observation scheduler was shown to scale to the full problem, producing observation schedules complete with all necessary enabling activities such as instrument reconfiguration, telescope repointing, data communication, etc. in a time frame acceptable for actual application. [MSCD92]. Complementary results demonstrated the ability of "multi-perspective" scheduling techniques to produce better quality schedules, in terms of balancing conflicting mission objectives, than a variant of the short-term scheduling algorithm currently being used in HST mission operations [SP92].

HSTS has subsequently been applied in other complex scheduling domains. It has been applied to the problem of planning the development of a new air base to support crisis-action personnel and equipment deployment [FM92]. Most recently, it has been used to develop of scheduler for actual

application to a second orbiting telescope, the Small Wave SubMillimeter Astronomy Satellite (SWAS), currently due to be launched in early 1995. [MS93].

4 Constraint Posting Scheduling

As indicated earlier, research in constraint-based scheduling has typically formulated the problem as one of finding a consistent assignment of start times for each goal activity. The HSTS framework, in contrast, advocates a problem formulation more akin to least-commitment planning frameworks: the problem is most naturally treated as one of posting sufficient additional precedence constraints between pairs of activities contending for the same resource so as to ensure feasibility with respect to time and capacity constraints. Solutions generated in this way typically represent a set of feasible schedules (i.e., the sets of activity start times that remain consistent with posted sequencing constraints), as opposed to a single assignment of start times.

While frameworks such as HSTS do not prohibit the use of "fixed time" scheduling techniques, there are several potential advantages to a solution approach that retains solution flexibility as problem constraints permit. From the standpoint of solution use, the generation of sets of feasible schedules provides a measure of robustness against executional uncertainty, allowing determination of actual start times to be delayed and minimizing the need for solution revision. From the standpoint of solution development, a constraint posting formulation of the problem can provide a more convenient search space in which to operate. During schedule generation, alternatives are not unnecessarily pruned by the need to (over) commitment on specific start times. When the need for schedule revision becomes apparent, modifications can often be made much more directly and efficiently through simple adjustment of posted constraints.

Given these potential advantages, much of our recent research has focused on development and evaluation of constraint-posting scheduling techniques. One line of research, generalizing directly from previous work in opportunistic scheduling but without the "fixed times" assumption, has led to development of a procedure called Conflict Partition Scheduling (CPS) [Mus93c]. CPS utilizes previously developed techniques for estimating resource contention[MS87] and is driven by recognition of resource capacity bottlenecks - time periods where there is contention among goal activities for the same resource capacity. CPS proceeds by iteratively identifying resource capacity bottlenecks (which is accomplished through use of a stochastic simulation technique) and acting to lessen the level of contention by posting ordering constraints among the activities competing for capacity at the most severe bottleneck. The iterative process continues until no capacity conflicts remain, at which point a final schedule has been determined. The search is simply restarted in the event that an infeasible solution state is reached. An experimental analysis of the performance of CPS on a set of benchmark constraint satisfaction scheduling problems demonstrated superior problem solving performance to two currently dominant "fixed-times" scheduling approaches - micro-opportunistic scheduling [Sad91] and min-conflict iterative repair [MJPL92]. The reader is referred to [Mus93c] for details. More recent work on CPS has aimed at the evaluation of different alternative CPS configurations (e.g., micro vs macro decision making, focused on capacity conflicts vs randomly focused) to establish the relative importance of different steps of the procedure and the performance trade-offs [Mus93a].

A second line of research has investigated the possibility of simpler, computationally cheaper alternatives to contention-based problem analysis when a constraint posting framework is assumed,

leading to development of a procedure called Precedence Constraint Posting (PCP) [SC93]. PCP couples the use of previously developed dominance conditions for incremental pruning of the set of feasible sequencing alternatives[EV76] with a simpler look-ahead analysis of the temporal flexibility associated with different sequencing decisions. At each step of the search, a measure of "residual temporal slack" is computed for each sequencing decision that remains to be made; the decision with the smallest residual slack is chosen as the most critical, and a precedence constraint is posted in the direction that retains the most flexibility. Whenever posting a constraint leaves other sequencing decisions with only a single feasible ordering, these unconditional decisions are also taken (i.e. the implied precedence constraints are also posted) before recomputing estimates of residual slack. Unlike CPS, which posts constraints only until all resource contention has been resolved, the PCP procedure terminates when either all pairs of activities contending for the same resource have been sequenced, or an infeasible state has been reached. Experimental results with PCP on the same suite of constraint satisfaction scheduling problems have shown comparable problem solving performance to contention-based scheduling approaches with orders of magnitude reduction in computational time[SC93].

More recent work with PCP has examined its use in more frequently encountered, optimization-based scheduling contexts (i.e., where the goal is not simply a feasible solution but a feasible solution that minimizes/maximizes some objective criterion). We are exploring two general approaches to adapting PCP for this purpose:

- *discrete relaxation search*, where PCP is embedded as a solution feasibility evaluator within a larger search through a space of possible constraint relaxations defined by the objective criteria, and
- *upper-bound improvement search*, where the PCP procedure itself is modified to directly incorporate the objective criteria (e.g., using estimates of "residual tardiness cost" as opposed to residual temporal slack), and a dynamically adjusted upper-bound solution provides the basis for search space pruning.

The utility of each of these approaches depends on characteristics of the specific optimization criterion that is considered. For example, the common manufacturing problem of minimizing weighted tardiness is better formulated as an improvement search, since there is no structure to support an effective search through the possible due date relaxations of all jobs. One criteria that is straightforwardly formulated as discrete relaxation search, however, is minimizing makespan (or overall duration of the schedule). We have developed a procedure, referred to as MULTI-PCP, which first establishes lower and upper bounds on the overall completion time of the schedule (using a critical path method and a simple dispatch heuristic respectively), and then searches for the minimum feasible "common due date" by repeatedly applying PCP to various dates within these bounds. We have contrasted the performance of this procedure with that of the shifting bottleneck procedure [ABZ88] (the current standard with respect to heuristic solutions to the makespan problem) on a set of previously studied benchmark problems. In these experiments, MULTI-PCP was shown to produce competitive solutions (more often than not closer to the optimum than the solutions obtained with the shifting bottleneck procedure) in equivalent or less computation time [CS93].

Finally, we are investigating the utility of constraint posting approaches in the context of scheduling experiments for an automated robotic chemistry workstation constructed within the Chemistry Department at CMU. This problem differs significantly in character from the benchmark manufacturing scheduling problems discussed above. In particular, experiment plans are characterized

by an initial set of activities which initiate the desired chemical reaction, followed by a sequence of sampling activities to monitor the progress of the reaction. In all cases, there are absolute temporal separation constraints between activities (e.g., consecutive sampling activities must be minimally separated by an hour and not more than 2 hours). The presence of separation constraints in all goal activity sequences diminishes the effectiveness of techniques which focus on individual activities at each step, and calls instead for "macro" scheduling techniques which simultaneously consider all activities of a given job (experiment), and incrementally add unscheduled jobs into the schedule. This is, in fact, the manner in which the "fixed times" scheduler currently used to drive the workstation operates. To consider the utility of a constraint posting formulation in this context, we implemented a variant of the current scheduler that differed only in that it posted sequencing constraints between activities (as opposed to of assigning activity start times) when scheduling a selected experiment. Comparative performance analysis over a range of realistic workstation loads indicated that this change alone yielded a 10-20% improvement in overall utilization of the facility, with a corresponding reduction in overall schedule makespan. Further details of this work can be found in [AS93].

5 Issues in Integrating Planning and Scheduling

The work summarized above indicates the promise of constraint posting approaches to scheduling, which co-exist naturally with the assumptions of temporal planning frameworks and promote the development of integrated planning and scheduling capabilities. We conclude by briefly identifying a few important open issues with respect to integration of scheduling and planning processes, and the further investigation of constraint-posting scheduling approaches:

- Opportunistic interleaving of goal sequencing and goal expansion - Our work to date in integrating planning and scheduling (e.g., in the HST domain) has assumed an integration framework where sequencing (scheduling) decisions drive the overall problem solving process; each goal sequencing decision defines a localized goal expansion (planning) subproblem and solution of this subproblem in turn provides more precise timing constraints to focus subsequent sequencing choices. Depending on problem characteristics, this integration assumption may be more or less valid. What is desired more generally are procedures which dynamically interleave expansion and sequencing decisions as a function of the perceived criticality of each in productively furthering the search.

- Dealing with aggregate capacity representations - One difficult problem in constraint posting scheduling is treatment of aggregate representations of shared resource capacity (typically a necessity in large-scale domains). Since constraint posting approaches avoid contention by posting precedence constraints among competing activities, consistent simultaneous allocation of aggregate capacity to multiple activities implies that activities must be implicitly allocated to specific atomic resources summarized by the aggregate resource (i.e., parallel sequences must be configured). This is the approach taken in [FM92]. But, such rigorous consistency enforcement defeats much of the advantage of introducing the abstraction in the first place.

- Reactive scheduling - The more flexible schedules produced by constraint posting schedulers appear to offer inherent advantages in managing executional uncertainty, in responding to unexpected events that invalidate aspects of the solution, and in incrementally revising schedules to accommodate changing goals and requirements. However, mechanisms for exploiting

these potential advantages and their relationship to the now substantial body of reactive scheduling techniques developed under "fixed times" representational assumptions remain largely unexplored.

- System generality versus system performance - Much of the work in scheduling and planning system development has emphasized generality (i.e. the ability to handle a larger class of problems or a broader set of constraints). This was a principal objective, for example, in the design of HSTS. However, the design goal of generality is quite often at odds with system performance objectives. HSTS, with its emphasis on structured domain models that exploit problem decomposability, pays some attention to this issue. But, at a broader level, it is often the case that specific types of constraints tend to dominate in particular domains. Not only are more general functional capabilities unnecessary in these cases, but their unconditional support at the architectural level is often the obstacle to meeting performance requirements. Greater emphasis needs to be placed on the development of system architectures that directly promote specialization (or customization) of component functionality for use in various application domains [SL93].

Acknowledgements: As implied by the references, Nicola Muscettola (now at NASA Ames) has played a central role in much of the work summarized in this paper, and I am grateful for the research collaboration we have had over the past several years. Acknowledgements are due also to Casper Cheng, who has been a driving force in the development of the PCP family of scheduling procedures.

References

[ABZ88] J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34(3):391–401, 1988.

[AK83] J. Allen and J.A. Koomen. Planning using a temporal world model. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 741–747, 1983.

[AS93] R.J. Aarts and S.F. Smith. A high performance scheduler for an automated chemistry workstation. Working paper, The Robotics Inst., Carnegie Mellon Univ., Sept. 1993.

[Bak74] K.R. Baker. *Introduction to Sequencing and Scheduling.* John Wiley and Sons, New York, 1974.

[CS93] C. Cheng and S.F. Smith. A new algorithm for job shop scheduling to minimize makespan. Working paper, The Robotics Institute, Carnegie Mellon Univ., Aug 1993.

[DM87] T.L. Dean and D.V. McDermott. Temporal data base management. *Artificial Intelligence*, 32:1–55, 1987.

[EV76] F. Roubellat Erschler, J. and J.P. Vernhes. Finding some essential characteristics of the feasible solutions for a scheduling problem. *Operations Research*, 24:772–782, 1976.

[FHN72] R.E. Fikes, P.E. Hart, and N.J. Nilsson. Learning and executing generalized robot plans. *Artificial Intelligence*, 3:251–288, 1972.

[FM92] R.E. Frederking and N. Muscettola. Temporal planning for transportation planning and scheduling. In *Proc. IEEE Conf. on Robotics and Automation*, April 1992.

[Lan88] A. Lansky. Localized event-based reasoning for multiagent domains. *Computational Intelligence*, 4:319–340, 1988.

[MJPL92] S. Minton, M. D. Johnston, A. B. Philips, and P. Laird. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58:361–205, 1992.

[MS87] N. Muscettola and S.F. Smith. A probabilistic framework for resource-constrained multi-agent planning. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 1063–1066. Morgan Kaufmann, 1987.

[MS93] N. Muscettola and S.F. Smith. Constraint-directed integration of planning and scheduling for space-based observatory management. In *Proceedings SOAR-93*, August 1993.

[MSCD92] N. Muscettola, S.F. Smith, A. Cesta, and D. D'Aloisi. Coordinating space telescope operations in an integrated planning and scheduling architecture. *IEEE Control Systems Magazine*, 12(1), February 1992.

[Mus93a] N. Muscettola. An experimental analysis of bottleneck-centered opportunistic scheduling. Technical Report CMU-RI-TR-93-06, The Robotics Institute, Carnegie Mellon University, March 1993.

[Mus93b] N. Muscettola. Hsts: Integrating planning and scheduling. Technical Report CMU-RI-TR-93-05, The Robotics Institute, Carnegie Mellon University, March 1993.

[Mus93c] N. Muscettola. Scheduling by iterative partition of bottleneck conflicts. In *Proceedings 9th IEEE Conference on AI Applications*, March 1993.

[Sad91] N. Sadeh. *Look-ahead Techniques for Micro-opportunistic Job Shop Scheduling*. PhD thesis, School of Computer Science, Carnegie Mellon University, March 1991.

[SC93] S.F. Smith and C. Cheng. Slack-based heuristics for constraint satisfaction scheduling. In *Proceedings AAAI-93*, Washington DC, July 1993.

[SL93] S.F. Smith and O. Lassila. Reconfigurable systems for reactive production management. In *Proceedings IFIP TC5/WG5.7 International Workshop on Knowledge-Based Reactive Scheduling*, pages 91–104, Athens, Greece, October 1993.

[SOMM90] S.F. Smith, J.Y. Ow, P.S. Potvin, N. Muscettola, and D. Matthys. An integrated framework for generating and revising factory schedules. *Journal of the Operational Research Society*, 41(6):539–552, 1990.

[SP92] S.F Smith and D.K. Pathak. Balancing antagonistic time and resource utilization constraints in over-subscribed scheduling problems. In *Proceedings 8th IEEE Conference on AI Applications*, March 1992.

[Ver83] S. Vere. Planning in time: Windows and durations for activities and goals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5, 1983.

[ZDG90] M. Zweben, M. Deale, and R. Gargan. Anytime rescheduling. In *Proc. DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*. Morgan Kaufmann Pub., Nov. 1990.