Planning Tactics within Scheduling Problems

Stephen F. Smith and **Terry L. Zimmerman**

The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213
{sfs, wizim}@cs.cmu.edu

Abstract

In this paper we consider the possibilities and potential advantages for exploiting automated planning techniques in the service of solving scheduling problems. The core competency of scheduling technologies is allocation of resources to pre-specified networks of competing activities (typically belonging to multiple processes) to maximize aspects of global system behavior. However, in practical domains it is rarely the case that the problem can be treated strictly as an allocation problem (i.e., strictly as a problem of enforcing disjunctive resource constraints in conjunction with specified temporal constraints). Rather some level of dynamic action selection is invariably required, typically to transition resources from one usage to the next. To retain scalability, schedulers tend to make use of locally circumscribed assumptions about the dynamics of resource usage that fit the problem at hand, which allows for efficient generation of resource-support plans without explicit reasoning about goals. But, these approaches can be overly restrictive in many cases, and they also tend to be difficult to extend and reuse.

1. Introduction

Within the A.I. community, the fields of automated planning and scheduling focus on solving complementary problems. Planning is concerned with determining *what to do* – i.e., synthesizing networks of actions that achieve a certain goal (or set of goals) given a specific initial state (or set of initial states). Scheduling is concerned with determining *when and how* – i.e., which resources should be allocated to various actions and in what order (or at what time) so as to maximize overall performance in some manner. These respective emphases, of goal-directed action selection on one hand and resource allocation over time on the other, lead to different types of problem formulations. The prototypical planning problem can be characterized as configuring the actions of a single agent to achieve specified goals, using general propositional descriptions of the enabling conditions and consequent effects of various actions. A prototypical scheduling problem, centers on reconciling competing requests for shared resources by multiple pre-specified processes, using more-specialized descriptions of resource requirements and resource availability over time. These different formulations and orientations naturally have led to different search procedures and solution techniques.

Despite the broad academic distinctions between planning and scheduling, practical problems are rarely so neatly decomposable. In generating a process plan in a manufacturing planning context, for example, the desirability of different process alternatives often depends critically on availability of the resources required, and failure to consider current production conditions and commitments can lead to inefficient (re)planning cycles. Likewise, the allocation of transport vehicles to service multiple requests in a logistics scheduling context often requires the intermediate generation of auxiliary enabling actions such as repositioning the vehicle, crew rest, etc., which cannot be pre-planned in a context independent manner.

Recognition of these inter-dependencies has led researchers in the planning and scheduling communities to focus increasingly in recent years on more-integrated solutions [Muscettola et al. 1992 Smith 1993, Hildum et al. 97, Rabideau et al. 99, Jonsson et al. 2000, Smith et al. 2000, Myers et al. 2001, Srivastava et al. 2001]. Coming at it from a planning perspective, frameworks for action selection have been pushed beyond classical assumptions to encompass scheduling sorts of constraints (e.g., metric, temporal and resource constraints), as evidenced by the evolution of the AIPS/ICAPS planning competition (Fox and Long, 2003). The progress made in this regard improves possibilities for exploiting planning techniques in problems that involve scheduling.

The argument for leveraging strengths of both scheduling and planning processes remains compelling however, and the manner in which these processes might be inter-leaved remains an interesting question. The perspective most frequently advanced assumes scheduling to be an adjunct to (or subsidiary component of) a broader planning process. While reasonable at a theoretical level, this perspective ignores several practical issues:

Problem fit - Many real world problems are resource-centered, and reasoning about resources and time is the crux of the scheduling problem. These problems are naturally formulated as scheduling problems even though their solution requires some degree of dynamic action selection.

Knowledge demands - It is certainly the case that the set of actions provided as input to a resource-centered problem (i.e., the processes that are competing for shared resources) are generated by some upstream planning process.

However, in many practical domains such planning processes (e.g., how to make a part in manufacturing; what itinerary to fly to accomplish a given airlift mission) tend to be ill-structured and knowledge-intensive, and automation is not a tractable alternative.

Human bias – Partly due to the above-mentioned complications with automating the process, partly due to the relative inflexibility of planning techniques with respect to incorporation of user decisions and advice, and partly due to the fact that users feel ownership of this process, users generally prefer to do the high-level planning (choosing which actions to do) themselves.

Taking these points as a starting point, we argue for the reverse integration perspective in this paper – that of embedding planning capabilities within a master scheduling process. To motivate this position, we take a retrospective look at some of scheduling applications we have tackled in the past from the perspective of dynamic action selection. One fact that becomes immediately clear is that practical scheduling problems rarely can be cast as problems in which action selection (or planning) can be completely factored out and assumed to be a pre-cursor activity. Rather, determination of the feasibility of various resource allocation decisions generally requires the (dynamic) establishment of an appropriate set of enabling "resource support" actions. We have previously solved these sorts of embedded planning problems by making problem-specific assumptions and employing a predetermined set of plan templates within the scheduler's search process. However, both flexibility issues and the desire for a solution approach that is more readily adaptable to other problems can argue for incorporating more robust planning technology. We outline here conditions under which it may be advantageous to employ automated planning as an adjunct to scheduling and discuss some possibilities.

2. The Case for Planning in Scheduling

To expand the succinct definition given in the introduction, scheduling can be viewed as the problem of assigning limited resources to tasks (activities) over time to optimize one or more objectives. In a scheduling problem, input demands provide top-level goals (e.g., produce 10 widgets by next Friday, transport package *P* from location *A* to location *B* by Thursday), and one or more activities (typically known at the outset) must be executed to satisfy a given demand. Activities require resources to execute and the scheduler must reason about which of a set of available resources should be assigned to perform each activity. These allocation decisions may be subject to an arbitrarily complex variety of constraints, including precedence relations, resource capacity, activity performance duration, and other auxiliary conditions (sub-goals) entailed by the assignment of a given resource. It is the presence of this last aspect in particular that compels us to consider exploiting planning methods in the service of scheduling.

To provide a context for discussing this class of scheduling problems, we first provide a basic schema of the search a scheduler might carry out. We then consider a progression of practical scheduling problems that require increasingly more flexible dynamic action selection capabilities, pointing out the restrictive nature of the approach we have taken in some cases to solve them.

2.1 A Basic Scheduling Search Model

Figure 1 graphically depicts a core search (sub) procedure for adding an individual activity to a schedule that highlights the essential decisions a scheduler must make. Given a specification of activity time restrictions (e.g., t_1 and t_2 , earliest and latest time bounds on execution) and resource requirements (e.g., what types of resources, R_{type},, can support the activity), the search considers alternative resources that might be assigned and for each, a set of intervals consistent with the activity's time restrictions during which the resource is currently available. Choices are evaluated according some performance criteria (e.g., finish as soon as possible) and an assignment is made.

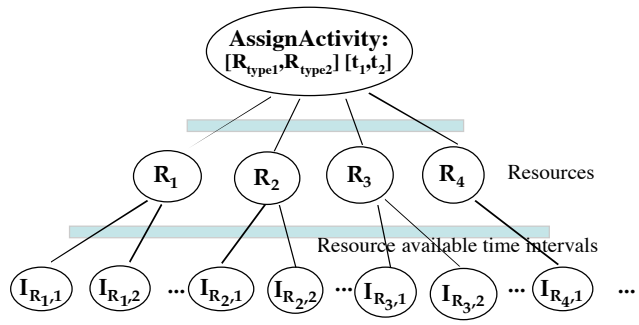


Figure 1: Scheduling an Activity

A complete scheduling procedure can be obtained by combining the use of this sub-procedure with a variable ordering heuristic, which dictates the order in which input activities to be scheduled are added into the schedule (e.g., [Becker and Smith 2000]). This greedy procedure can then be embedded in a larger optimizing search by either (1) randomizing the heuristic used (e.g., [Cicerello and Smith 2002]) or otherwise dynamically reprioritizing the variable ordering (e.g., Joslin and Clement 1998) and re-invoking it repeated times, or (2) using the solution obtained by the greedy generator as a starting point for further local search (e.g., [Kramer and Smith 2003, Syswerda 91]).

2.2 A Progression of Problems involving Dynamic Action Selection

It is the disjunctive nature of resource constraints (i.e., the fact that a resource can only support one activity at a time in the simplest case) that makes the scheduling problem hard. In a classical scheduling problem formulation, a resource is assumed to transition from unavailable to available (and vice versa) instantaneously. A resource *r* that is assigned to an activity *A* is unavailable to all

activities other than A from precisely the start of A to its end, and r can be immediately assigned to another activity B at any time point before or after the execution of A. Under such formulations, a simple accounting of resource usage over time (e.g., as depicted in Figure 2) is sufficient to determine the set of intervals $\{I_{r,1} \ldots\}$ during which resource r is available for assignment to some unscheduled activity.

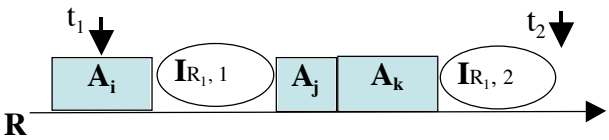


Figure 2: Resource availability profile

In most practical domains, however, specific auxiliary actions must be taken to transition a resource from one usage to the next, and determination of resource availability intervals is more complex. In manufacturing applications, a machine will often have to be "set up" to process a particular lot (e.g., specific fixturing may have to be attached), and then "torn down" and "set up" again to process the next lot in its queue. In some cases, such actions can be conveniently abstracted as a *state-dependent* duration computation. For example, the concept of a setup matrix, which specifies the time necessary to transition between any two configurations of a given resource, is commonly employed in manufacturing scheduling applications to instantiate necessary setup actions in the schedule. This leads to sequence dependent setup durations as shown in Figure 3.

Figure 3: Dynamic instantiation of "Setup" Actions

In other problem domains, however, determination of resource setup time (and hence computation of resource availability intervals) can be a much more challenging action selection problem. Generation of short-term observation schedules for the Hubble space telescope is an extreme example of an application of this nature [Muscettola et al. 1992]. Overall, this problem can be seen as a single-machine scheduling problem; except for selected serendipitous opportunities to do parallel science with a second instrument, the telescope generally performs one observation at a time. Yet, several inter-dependent threads of actions must typically be executed to transition from one observation to the next. If the next observation lies in a different direction, both slewing the telescope and target acquisition must be executed. If the instrument and instrument state required by an observation differs from the one preceding it, appropriate sequences of instrument cool-down and warm-up actions must be executed (with global restrictions on overall power usage). If the science tape recorder lacks sufficient space for data capture, data down-linking actions must be executed.

It is possible to construct abstractions of these enabling processes as dynamic duration computations for use in heuristics to select and order observations [Smith and Pathak 1994]. However, state-dependent interactions between elements of various telescope reconfiguration threads make it difficult to guarantee the accuracy of these computations. Hence, our solution to this problem interleaved use of abstract and detailed models [Muscettola, et.al. 1992]. The abstract model was used to determine which observation to schedule next, while invocation of a temporal planner (HSTS,). served to flesh out the full network of enabling actions for that observation, confirm feasibility with respect to associated time constraints (e.g., orbit windows) and establish precise duration information for use in "goal" (observation) selection in subsequent cycles. This system gives one example of the type of "planning within scheduling" architecture we are promoting, albeit one where the master scheduling/allocation problem involves only a single resource.

Location is another dynamic aspect of state that complicates determination of resource availability windows and has dominated many of the applications we have focused on in recent years. In transportation and logistics scheduling applications, the resources of interest are mobile, and assignment of a resource to a particular activity typically implies a change of location. Consider the problem of allocating aircraft to airlift missions over time [Becker and Smith 2000, Smith et al.2004]. An airlift mission in the simplest case defines a flight from a pickup location A to a drop off location B. Assignment of a particular aircraft to this activity correspondingly requires that the aircraft be present at location A at the start of the activity and at the end of the flight the aircraft will be at B. If not already at location A, it will be necessary to instantiate actions that "position" the aircraft for this flight. Likewise, if it is not possible or preferable to leave the aircraft at location B, the instantiation of appropriate "de-positioning" actions will be necessary. Like resource reconfiguration, the complexity of the supporting actions that must be dynamically derived to manage resource location may vary in different applications.

In many practical transportation domains, resource assignment entails management of both resource location and resource setup actions. In the USAF Air Mobility Command (AMC) scheduling application [Becker and Smith 2000, Smith et al 2004] the problem is short-term allocation of aircraft and aircrews to airlift and tanker missions. Over the two to three week horizon of interest, AMC flies several thousand air missions, using several hundred aircraft and a comparable number of air crews. An airlift mission specifies an itinerary of one or more consecutive flights (or legs), from an initial onload location to a final offload location. A tanker request, alternatively, specifies a refueling track (location), an "on-station" time, and a fuel off-load amount. Aircraft and air crews are organized into air wings, each associated with a particular home base to which they will normally return at the

completion of each mission. In order to allocate one or more aircraft and air crews from a particular wing to a given mission, it is necessary on the one hand to perform positioning (and de-positioning) actions and on the other, to ensure resources are appropriately "configured" for use (e.g., that crew rest time is budgeted in accordance with crew duty day constraints, that appropriate pre-mission procedures have been carried out, etc.). The overall goal is to maximize the number of missions that can be supported, subject to the relative priorities of competing missions.

Given the relative independence of different air missions, dynamic HTN-style expansion of positioning and de-positioning sequences (with standard velocity*distance duration computations) provides a reasonable default basis for evaluating alternative resource assignments. However, the problem of planning to make resources available for use can get considerably more complex. Though missions are planned such that aircraft and crews make round-trips by default, it is desirable where possible (within crew scheduled return date constraints, etc.) to take advantage of opportunities to combine complementary missions (e.g., transport cargo from *a* to *b*; re-position to nearby *c*; transport cargo from *c* to *d* which is close to home), and further optimize resource usage. Furthermore, although missions are typically supported by aircraft and air crews from the same base, it is sometimes necessary to stage crews from other nearby locations, or alternatively to swap in a new crew part way through the mission to avoid crew rest delays. There can also be secondary resource constraints that affect aircraft usage. At some locations there are significant "maximum-on-ground" (MOG) constraints, which can restrict travel times and increase overall mission duration. As the number of possible contingencies increases, so does the difficulty of anticipating and pre-specifying appropriate expansions.

Another complexity arises from the desire to exploit opportunities for synergy in the resource support actions that are introduced by different missions into the schedule. In the AMC domain distinct tanker missions are generally instantiated and scheduled to support different airlift refueling requests. Yet, depending on the amount of fuel required and the size of the tanker allocated, it may be possible for a single tanker mission to service multiple requests. In a related "air campaign scheduling" domain that we have also considered [Myers and Smith 2001], similar possibilities for synergy exist. For example, individual strike missions may each require supporting actions relating to radar-jamming. But the introduction of any given jamming mission into the plan will cover a distinct geographic area, and satisfy the resource usage requirements of all strike missions moving through that region.

The use of simple HTN-style templates allows scalable management of resource support plans in support of the scheduler's search.[1] However, this scalability is obtained at some loss of flexibility. The definition of resource support plan templates requires commitment to a (usually problem specific) set of assumptions about the dynamics of resource usage which allow templates to be instantiated without explicit reasoning about the state conditions (goals) that support plans are intended to achieve during the search for availability intervals. For example, a simple aircraft positioning template might assume that an aircraft and air crew must be allocated from the same location, in which case a single positioning flight sequence will cover all circumstances. As additional constraints introduce conditionality and additional context sensitivity to possible resource support actions, it becomes increasingly complex to define appropriate templates, and generally some customization or re-engineering of the scheduler's search operators becomes necessary. We argue next that more explicit forms of A.I. planning, may offer a better flexibility/performance tradeoff in some application contexts.

3. Incorporating Planning into scheduling

Restating and expanding the brief definition of planning given in the introduction: Automated planning can be described as the process of determining what actions (activities) to take and in what order so as to progress from an initial world state to a desired goal state. For a planning problem, besides the goals and the initial state, the input must define a set of domain operators, describing for each operator the set of effects produced when the action executes in a state containing certain preconditions. Planning focuses on making decisions about action (operator) choices, where choosing an action typically leads to a cascading series of entailed choices and these choices may interact in an arbitrary variety of complex ways). In the course of selecting actions automated planners must also deal with resource assignment via the same algorithm.

3.1 A planning problem within the AMC domain

The cascading series of subgoals that typifies a planning problem corresponds closely to the nature of the resource setup activities described in the previous section. Consider Figure 4, where we depict a simplified example of an aircraft positioning setup problem within the AMC domain as a partially expanded planning goal - action assignment graph. Here the scheduler requires two aircraft, a C-5 and a C-141, to be positioned to JFK no later than time 20 to support a given cargo airlift mission. (We simplify the depiction to show just lft for each event rather than a time window).

---

[1] In the AMC Allocator application, for example a schedule consisting of 1000 missions (~ 6000 activities) is generated in less than 5 seconds on a P4, 1GHz machine.

There are three air force bases within proximity of JFK airport from which the required aircraft might be assigned; bases B1 and B5 have both C-5 and C-141 aircraft but B2 has only C-141s. Each base has a single unit capacity resident ground crew and multiple flight crews, and a flight crew can only fly aircraft for which it is qualified. Preconditions for flying an aircraft include positioning a flight crew and ground crew preparation of the aircraft.

The shaded section of the Figure 4 graph roughly covers the sort of support activities that our current AMC scheduler models and would consider in attempting to allocate resources to the aircraft positioning tasks. If the resources needed to allocate an aircraft from a given base (aircraft, flight crew, and ground crew) are not available at that base within the time constraints, the search routine moves on to consider allocation from another base.

In reality though, this does not model the full range of options available to human planners at AMC. If a required aircraft is available at a given base but a flight crew is not the human planners can and do consider options for transporting a crew from a base that has an extra crew available (e.g. due to aircraft being out for maintenance). The possibilities for borrowing a flight crew to operate a C-5 out of base B1 are logically represented in the un-shaded portion of Figure 4. We have posited three transport options; flying a crew from a neighboring base on a dedicated shuttle, using ground transport, or putting the crew on a previously scheduled flight from B5 to B1 (i.e. one associated with another mission). The dedicated shuttle is fastest but requires an available shuttle aircraft and the B5 ground crew to ready it, the ground transport option is the slowest, and the option for piggybacking the crew on an existing flight between the bases requires that the flight arrives in time for the C-5 mission. Each of these options entails constraints that impact feasibility.

The increasing interdependency and complexity entailed by this more realistic model of aircraft positioning motivates adoption of automated planning methods. Viewed as a planning problem, the Figure 4 goals consist of the two propositions specifying the particular aircraft types needed at JFK at lft=20. The initial state consists of all the current and relevant resource availability profiles and the domain actions can be defined as depicted in Figure 4. During the planning search process, the truth values associated with Figure 4 leaf nodes (shown as

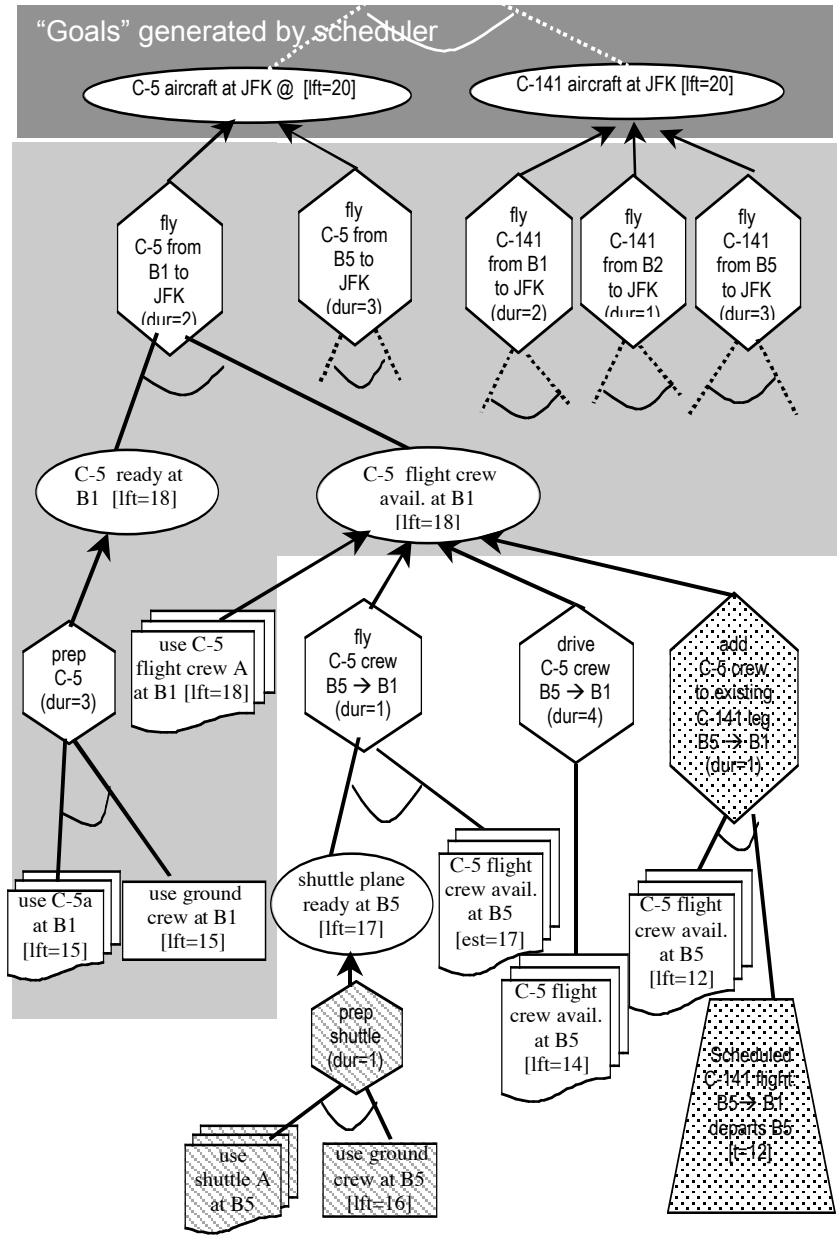
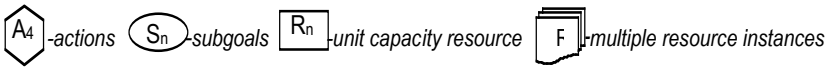


Figure 4. Depiction of planning subgoals and action assignment options for a portion of an AMC domain problem. In the directed graph subgoal nodes appear as ellipses and incoming edges represent alternative satisfying actions (hexagons). Each action has edges leading to one or more preconditions that all must be satisfied for the action to be applied. The shaded portion of the figure roughly represents positioning logic currently modeled in the AMC scheduler. Example negatively interacting actions are shown in hatching and a positively interacting action appears as speckled elements.



rectangles) are determined via checking the given resource's availability profile. If the desired resource is available given the associated constraints, the resource must be temporarily allocated until either a valid plan is found or the assignment is backtracked over.

Even in this simple example it becomes unwieldy to apply a priori, HTN-style templates of the many possible cascading resource assignment sequences. There is a growing number of possibilities for satisfying the subgoals that are conditional on the presence or absence of state variables. Situations such as this argue for more explicit planning methods that can reason directly about goal satisfaction and the wide potential variety of goal interactions. More generally, the case for employing automated planning as an adjunct to scheduling becomes more compelling as any of the following parameters of a particular problem *increase*:

- The number of entailed conditions (subgoals) that must be satisfied by additional non-trivial activities in order to consistently allocate a task on the scheduling agenda.
- The number of state variables (beyond the standard variables associated with resource profiles)
- The number of options for establishing a subgoal condition
- The number of interactions, negative and positive, amongst sets of subgoals and the actions that achieve them.

In problems where these parameters have relatively high values, attempting to generate and validate candidate resource assignments via pre-specified templates becomes increasingly unwieldy. Moreover, the current generation of time and resource contention-based heuristics that lie at the heart of a successful greedy search within scheduling are apt to be misled as the complexity reflected by these factors increases.

One key concern when contemplating the incorporation of planning within a scheduler core search process is scalability. For problems as large-scale as AMC, for example, thousands of planning sub-problems must be solved in the course of generating a schedule. However, resource support planning problems of the sort depicted in Figure 4 tend to be relatively contained and appear to be small by current temporal planning standards. In fact, for the type of logistics domain we have used as a motivating example the recent AIPS/ICAPS planning competitions have demonstrated basic viability.

3.2 Roles for planning assistance to the scheduler

There are a number of possible ways in which planning technology might be exploited to enhance a scheduler's ability to manage resource usage constraints. We discuss some of the most promising ones here.

Planning to satisfy resource usage requirements. As suggested above, planning can serve to generate in a valid and consistent manner, all the support and setup activities entailed by the allocation of a resource to a high level activity. In relation to simple template instantiation approaches typically employed for this purpose, explicit goal-directed planning enables the specification and generation of a broader range of options (particularly in the presence of contingent and interacting constraints), while at the same time providing a flexible basis for model extension and reuse. Furthermore, by confining the planner's responsibility to the generation of those actions required to transition resources from one usage to the next, the domain model (knowledge) required by the planner can be kept to a manageable size.

We have discussed the current AMC practice in which the human planner might consider possibilities for borrowing crew from neighboring bases. Successfully consigning this workaround to an automated planner involves subtleties beyond what we have discussed above. In the Figure 4 example, the planner should consider borrowing only *spare* crews from another base, otherwise it could end up with a solution that has the C-5 being positioned from B1 with a crew transported from B5, when a shorter, less-costly solution would just use a C-5 and crew from B5. This can be assured via modest augmentations of the planning domain theory, but less straightforward is implementing the borrowing of any crews that end up with extra idle time *after* the scheduler allocates all missions that it can. Since this scheduling problem is typically over-subscribed, it may be possible in a post-processing step, to go back and allocate missions that could not be assigned in the first pass by considering the transport of flight crews with idle time to locations where there are simultaneously available aircraft.

Communication and tracking of the reasons for planner failure in specific allocation contexts can provide a direct basis for implementing such multi-pass search strategies. Dependency directed backtracking (DDB) and explanation based learning (EBL) techniques have been successfully applied to both planning graph based planners [Kambhampati 2000] and partial order planners [Kambhampati et al. 1997], showing that concise sets of constraints responsible for planning failures can be compiled during search. With modest massaging, these 'conflict sets' should be able to inform the scheduler which constraints need to be worked around to schedule a given activity, and these conflicts sets can be revisited for any activity left unassigned after first-pass scheduling.

More generally, it may make sense for the scheduler to utilize the planner in an iterative deepening manner. If resources are not found to be available under normal operating assumptions (airplane and crew at same location) and flight crew availability is the reason for failure, the planner is re-invoked to seek longer plans involving transporting crews from other bases, etc.

Planning for "high quality" resource-support plans. Whereas the first step is to generate resource-support plans which establish that a given resource can be feasibly allocated to a given activity, a further goal (particularly in contexts where diverse options are possible) is to generate high quality support plans. As evidenced by recent planning competition domains, more recent planning

research has tackled the problem of finding plans that seek to optimize over multiple quality criteria, such as cost, fuel usage, and number of actions [Gerevini and Serina 2002, Zimmerman and Kambhampati 2002, Zimmerman 2003]. This capability may be exploitable within a scheduler by charging the automated planner with the task of finding higher quality solutions along multiple criteria for the sub-problems handed to it. Since generation of resource-support plans is a fairly constrained problem, it may even be feasible to quickly generate virtually all solutions to the planning problem and allow the scheduler to choose amongst them [Zimmerman 2003].

Planning to capitalize on positive interactions. Another dimension along which an explicit planning component can enhance the scheduler's search is in providing a flexible, principled basis for exploiting activities already in the schedule to further optimize resource utilization. Our current AMC scheduler provides a specialized "mission merging" capability designed to minimize nonproductive flying time; however there are many other circumstances where joint consideration of the activity currently being scheduled and other activities already in the schedule can lead to better shared use of resources.

An example of such “positive interaction” (i.e. an action that can achieve more than one plan goal) is depicted in Figure 4 by the action that places the needed flight crew aboard a previously scheduled flight between B5 and B1 (speckled hexagon and trapezoid). Consideration of such possibilities within the current scheduling algorithm would entail significant hardwired coding of search operators that must be modified for each situation where these interactions are of interest. There are also numerous instances of “negative interactions” (actions that can interfere with execution of other desired plan actions): For example, the option of shuttling a crew to B1 from B5 requires both a shuttle aircraft and the ground crew at B5, and these requirements may conflict with subsequent planning to satisfy the second goal (positioning the C-141).

A more ambitious role for the planning component is to have the scheduler pass it a set of goals that are entailed by the allocation requirements of a *set* of activities. The motivation here is to take a step away from the greedy, "allocate one activity at a time" search of the scheduler, and consider possibilities for synergistic resource usage up front. Since the planning process can directly deal with negative and positive interactions amongst these goals, the integrated scheduler/planner may generate a better quality solution. In the Figure 4 example, such a joint allocation process might result in the planner choosing to transport the crew via the more 'costly' shuttle aircraft rather than placing them aboard an existing flight (which may require more of their time due to its schedule), if opportunities exist to transport other crews associated with satisfying another goal during the shuttle return to home base.

A key problem for the scheduler in this mode is efficient identification of sets of goals/activities for which allocation as a group might be advantageous. The tradeoff here, of course, is possibly large increases in the required planning effort. In some cases, simple clustering criteria (e.g., geographic proximity, complementary cargo trajectories in the AMC domain) could offer good starting points. But whether a practical partitioning of goals created by the scheduler can be found is an open question.

Planning heuristics for ordering resource alternatives. The planning process we have portrayed for finding a resource support plan confirming a feasible resource allocation can also be viewed in a different light; as informing the scheduler regarding the best allocation choices. Referring back to the basic search schema in Figure 1, the first ply of the search generates a set of possible resource choices. In simpler cases the search might enumerate and explore all possibilities, but pragmatically, it often makes sense to employ heuristics to prioritize and search the space of alternatives in a "best first" fashion. Since the speed and complexity associated with transitioning a resource into a usable state (e.g. setup) relates directly to its viability as an allocation option, some degree of planning such transitions might well serve as an informed heuristic.

The variety of heuristic metrics that might be generated via planning methods have particular appeal as a source of heuristic guidance here because the associated space of 'computational effort versus level of informedness’ is large and varied. The range of possibilities extends from conducting full-blown metric temporal planning on each sub-problem to methods that conduct no search at all, such as mining a planning graph[2] for heuristic metrics. In the full search mode the planner can not only identify which candidate goal sets can be satisfied under current schedule constraints but can rank these goal sets on the basis of one or more quality measures of the plan solutions generated.

At the other end of the range, where actual planning is minimal, there has been extensive research associated with extracting powerful heuristics from the planning graph [Hoffmann, J 2000, Nguyen, X. et. al. 2001, Do and Kambhampati, 2002]. These so-called "distance-based heuristics" nominally provide an optimistic measure of the reachability of a state, but can also be used to estimate other attributes such as cost and time (given the appropriate propagation on the planning graph). The range of the computational effort vs. informedness space grows still further when considering alternatives for exploiting the planning graph: Computationally cheaper heuristics can be obtained by reducing effort spent on constraint propagation

[2] The planning graph is a data structure first introduced with Graphplan [Blum & Furst, 1997] that consists of alternating levels of planning domain actions and their propositional effects. Beginning with the set of propositions given as initially true, each subsequent proposition level compactly represents the states that *might* be reachable after application of a subset of the actions appearing in the previous action level. Mutex constraints between pairs of actions and pairs of propositions serve to bound the feasible states by implementing partial 2-consistency.

in the graph. One of the more heavily exploited planning heuristics, the "relaxed plan", can be generated in this manner. By ignoring negative interactions between actions, the relaxed plan idea has been widely used to quickly produce approximate (but not provably correct) plans for achieving candidate goal sets [Hoffmann, J 2000, Nguyen, X. et. al. 2001, Do and Kambhampati, 2002].

In the middle range of the computational effort vs. informedness space lie various reduced planning search options such as depth-limited or time-bound search. Depending on the sub-problem a sufficiently informed heuristic might be derived from something less difficult than full-blown metric temporal planning, such as classical planning with action costs.

4. Conclusions

In this paper we promoted the use of planning techniques to support solution of action selection sub-problems that arise within a scheduler's search for feasible resource assignments. We identified limitations of the template instantiation approach that we have previously used to handle resource configuration in various scheduling applications, and outlined a number of ways in which automated planning techniques might be used to overcome them. We are currently exploring the development of a scheduling/planning architecture based on these ideas.

Acknowledgements: The work reported in this paper has been supported in part by the U.S. Air Force Research Laboratory - Rome, under contract F30602-02-2-0149, by the National Science Foundation under contract # 9900298 and by the CMU Robotics Institute.

References

Becker, M. and S.F. Smith, "Mixed-Initiative Resource Management: The AMC Barrel Allocator", *Proc. 5th Intl. Conf. on AI Planning and Scheduling*, 2000.

Blum, A. and Furst, M.L. (1997). Fast planning through planning graph analysis. *Artificial Intelligence.* 90(1-2). 1997.

Cicirello, V. and S.F. Smith, "Amplification of Search Performance through Randomization of Heuristics", *Proc. 5th Int Conf on Principles of Constraint Programming*, 2002.

Do, Minh B. and Kambhampati, S. 2002. 'Planning Graph-based Heuristics for Cost-sensitive Temporal Planning'. In *Proc. 6th Intl. Conf. on AI Planning and Scheduling.* April, 2002.

Fox, M. and Long, D. (2003) "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains", *Journal of A.I. Research,* Volume 20, pages 61-124.

Gerevini, A. Serina, I., (2002). LPG: A planner based on local search for planning graphs with action costs. In *Proc. 6th Intl. Conf. on AI Planning and Scheduling.* April, 2002.

Hildum, , D.W. , N. Sadeh, T.J. Laliberty, J. McAíNulty, S.F. Smith and D. Kjenstad, "Blackboard Agents for Mixed-Initiative Management of Integrated Process Planning/ Production Scheduling Solutions Across the Supply Chain", Proc. 9th Conf. on Innovative Applications of AI, July, 1997

Hoffmann, J. 2000. The FF Planning System: Fast Plan Generation Through Heuristic Search In *JAIR-14*:p.253 - 302.

Jonsson, A., P. Morris, N. Muscettola, K. Rajan and B. Smith, "Planning in Interplanetary Space: Theory and practice", *Proc. 5th Intl. Conf. on AI Planning and Scheduling,* 2000.

Joslin, D. and D. Clement, "Squeeky Wheel Optimization", Proceedings AAAI-98, Madison WI, July 1998.

Kambhampati, S. Katukam, Y. Qu, Failure Driven Dynamic Search Control for Partial Order Planners: An explanation-based approach. *Artificial Intelligence 88(1-2), 1997*

Kambhampati, S. 2000. Planning Graph as a (dynamic) CSP: Exploiting EBL, DDB and other CSP search techniques in Graphplan. JAIR, *12*,1-34, 2000.

Kramer, L. and S.F. Smith, "Maximizing Flexibility: A Retraction Heuristic for Over-subscribed Scheduling Problems", *Proceedings of IJCAI-03*, August 2003.

Muscettola, N., S.F. Smith, A. Cesta and D. D'Aloisi, "Coordinating Space Telescope Operations in an Integrated Planning and Scheduling Framework", *IEEE Control Systems*, 12(1), 1992

Myers K.L, S.F. Smith, D.W. Hildum, P.A. Jarvis, R de Lacaze 2001. Integrating Planning and Scheduling through Adaptation of Resource Intensity Estimates", Proceedings ECP-01, Toledo, Spain, Sept. 2001.

Nguyen, X., Kambhampati, S., and Nigenda, R. 2001. Planning Graph as the Basis for deriving Heuristics for Plan Synthesis by State Space and CSP Search. *AIJ-135*:p.73-123.

Rabideau, G., R. Knight, S. Chien, A. Fukunaga, and A. Govindjee. "Iterative Repair Planning for Spacecraft Operations using the ASPEN System", *Proc. Intl. Symp. Of AI, Robotics and Automation for Space*, 1999.

Srivastava, B., Kambhampati, S. and Do, M. 2001. Planning the Project Management Way: Efficient Planning by Effective Integration of Causal and Resource Reasoning in RealPlan. *Artificial Intelligence Journal, 131 (p.73-134)*, 2001.

Smith, D.E., J. Frank and A.K. Jonsson, "Bridging the Gap Between Planning and Scheduling", *Knowledge Engineering Review*, 15 (1) 2000.

Smith, S.F. 1993. "Integrating Planning & Scheduling: Towards Effective Coordination in Resource-Constrained Domains", Proceedings Italian Planning Workshop, Rome Sept. 1993.

Smith, S.F. and D.K. Pathak, "Balancing Antagonistic Time and Resource Utilization Constraints in Oversubscribed Scheduling Problems", *Proc. 8th IEEE Conf. on Applications of Artificial Intelligence*, Monterey, CA, March, 1992

Smith, S.F., M.A. Becker and L Kramer, "Continuous Management of Airlift and Tanker Resources: A Constraint-Based Approach", *Mathematical and Computer Modeling: Special Issue on Defense Transportation*, To Appear, 2004.

Syswerda, G., "Schedule Optimization Using Genetic Algorithms", in (ed L. David) Handbook of Genetic Algorithms, Van Nostrand Reinhold, NY,1991.

Zimmerman, T., Kambhampati, S. 2002. Generating parallel plans satisfying multiple criteria in anytime fashion. In 6th Intl. Conf. on AI Planning and Scheduling, Workshop on Planning and Scheduling with Multiple Criteria,. April, 2002.

Zimmerman, T. 2003. Exploiting memory in the search for high quality plans on the planning graph. PhD dissertation, Arizona State University.