Board-laying Techniques Improve Local Search in Mixed Planning and Scheduling

Russell Knight, Gregg Rabideau, and Steve Chien
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109-8099
firstname.lastname@jpl.nasa.gov

Abstract

When searching the space of possible plans for combined planning and scheduling problems we often reach a local maximum and must either backtrack or otherwise modify the plan to make further progress. Occasionally, we can make large steps in the search space by aggregating constraints. Our techniques improve the performance of our planner/scheduler on real problems.

Introduction

This paper describes an approach that can improve search in the plan space. This is achieved by "jumping" within the search space by using an aggregation technique. Additional costs of the technique include the overhead involved in reasoning about the aggregation. Previous work in the form of macro operators and clustering addresses the problem of deciding on an appropriate collection of activities, heretofore referred as an aggregation. We apply a simple heuristic technique to gather the aggregation and reason about the aggregation as a single unit. We claim that our reasoning techniques for a given aggregation facilitate a sound and complete local search.

Board-Laying: an Analogy of Aggregation

Consider searching the space of plans, where a plan consists of a collection of activities possibly related to one another with temporal constraints as in Figure 1.

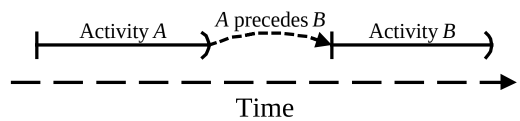

Figure 1 A plan consisting of two activities related to each other via a temporal constraint.

Consider that each activity also has resource and state constraints or *reservations* (Figure 2).

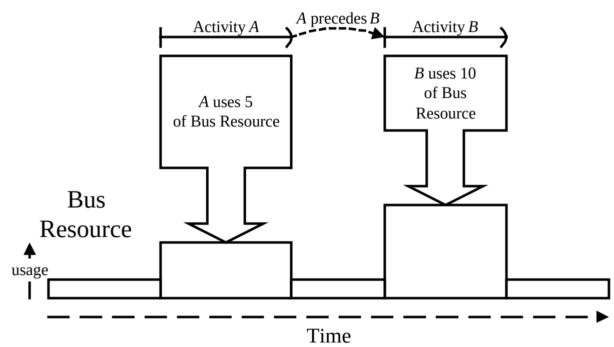

Figure 2 Reservations of activities on a shared resource

We consider a violation of either a temporal constraint or a reservation a *conflict*. If we start with a plan that contains conflicts of temporal constraints and reservations, we might decide to repair the temporal constraint conflicts and then the reservation conflicts. When we have repaired all of the temporal constraint conflicts, we have climbed a hill to a certain point, where hill climbing is seen as reducing conflicts. (Note: this is the dual of the "gradient descent" analogy in which valleys are more optimal than hilltops.) If we have no other temporal constraint conflicts, then we would attempt to resolve the reservation conflicts, but it may be that resolving a reservation conflict implies causing temporal constraint conflicts. In this case, we have reached a local maximum—in the sense of hill climbing, we have reached a hilltop, as in Figure 3.

One option is to simply descend the hill and solve the reservation conflicts, and then re-solve the temporal constraint conflicts. A planner that searches completely will have no trouble with this as because all options are considered, but most real domains we face have too many options, which leads to an impossibly long search. However, if we employ local search techniques, we may find ourselves descending and ascending the same hill, making no progress toward a global solution.

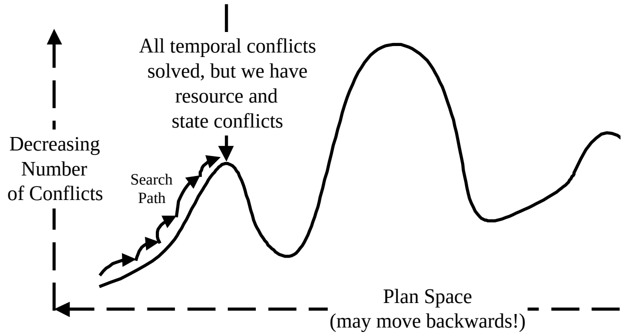

Figure 3 Reaching a hilltop in the search of the plan space

What we really want is to move from one hilltop to other local hilltops without descending. This is advantageous in that we know that we are not ascending the same hill and that we continue the search "up". One way of enabling this sort of operation is to reason about collections of activities instead of individual activities. This allows us to avoid some of the descending associated with reasoning about individual activities. In this sense, aggregation is equivalent to laying boards from one hilltop to nearby hilltops, as in Figure 4.

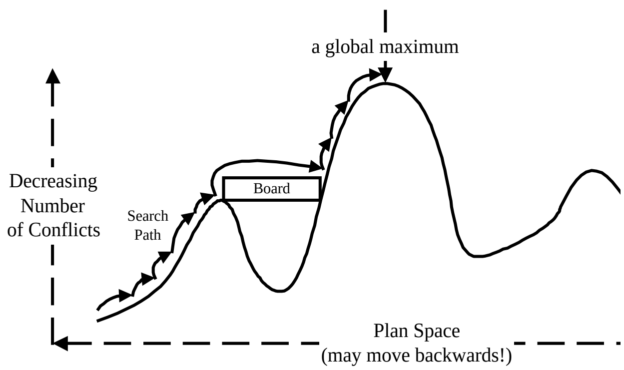

Figure 4 Avoiding backtracking while achieving a less-conflicting plan using board laying

Unfortunately, board-laying is not free. We must construct a board i.e. we must reason about aggregations of activities with respect to shared states and resources as well as temporal constraints.

Our criterion for inclusion into an aggregation is motivated by the structure of problems presented by modelers to our planner/scheduler: ASPEN (Rabideau, G., Knight R., Chien S., Fukunaga A., Govindjee A.). Often, a complex activity is modeled as a collection of simpler activities related to each other via temporal constraints. Therefore, we first choose an activity that has a reservation violation, and then we simply gather associated activities by calculating the connected component of the original activity in the temporal constraint network. This takes advantage of the intent of the modelers in that we assume that required activities to satisfy temporal constraints will need to be considered while satisfying resource and state constraints. Obviously, if this is not the case our aggregation criterion will not produce helpful collections. Therefore, we make no claims as to the appropriateness of our aggregation technique—we only claim that it is justifiable given the models actually produced for domains without respect to any particular solution technique.

To summarize the concept and issues of aggregation, we believe that reasoning about collections of activities (as opposed to individual activities) is helpful in finding a solution to a combined planning/scheduling problem. We believe this is due to the increased complexity introduced by the backtracking required for reasoning about individual activities. To avoid this, we will collect a group of activities that are related to each other with temporal constraints and then attempt to schedule the entire collection at once. In a sense, we assume that decisions concerning the temporal constraints among members of the collection will hold even when shifted "lock-step" in the schedule either forward or backward in time. Note that although research concerning flexible intervals between members of the collection might be fruitful, it is left as future work.

Given that we have chosen a collection of activities, does our technique: 1) improve on backtracking techniques and 2) improve search on any reasonable domain? Our empirical analysis compares our technique with a technique that performs no aggregation and thus must backtrack (answering 1), and we use domains from space-exploration (answering 2).

Reasoning about Collections of Activities

Consider the task of scheduling interdependent sets of activities in a combined scheduling/planning problem. This problem is an important aspect of solving combined planning and scheduling problems. In many approaches to combined planning and scheduling, one alternates between finding activities to satisfy pre- and post-conditions (planning) and finding temporal assignments and resources for those activities (scheduling). Complex activity placement is also an important component of many scheduling problems, as finding temporal assignments for complex activities can be computationally challenging.

This work advances the approach of moving collections of activities whose temporal relationships among themselves are fixed. We proceed by describing our motivation, defining the problem, and describing the solution. Finally, we present empirical evidence in favor of our technique.

We assume that we have already gathered activities into an aggregate, as described earlier. We explicitly represent interactions between the activities, or more specifically, interactions between the constraints on shared states and resources.

For example, consider a pair of activities that affect a battery (see Figure 5). The first activity a_1 uses 10 amp-minutes, while the second activity a_2 restores 10 amp-minutes. If we schedule a_1 individually, we find no intervals that will not cause an over-subscription of the battery, because activity a_3 has already fully depleted the battery by the end of the current schedule. But, if we schedule these together, we find placements that are valid. The positive effect a_2 has on the schedule makes up for a_1's usage. We wish to handle this sort of constraint-interaction for shared states and resources.

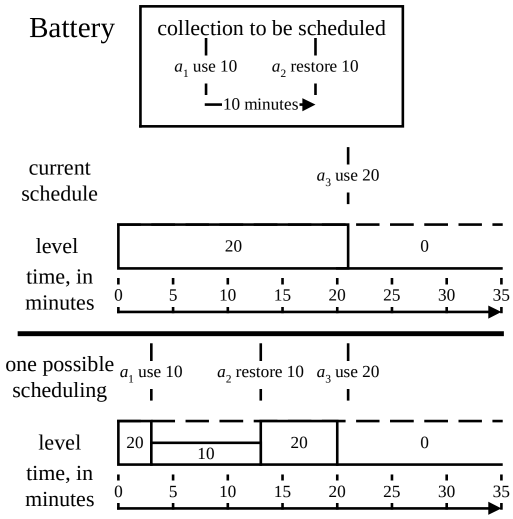

Figure 5 Battery interaction example

Empirical Evaluation

In our empirical analysis we use four models (and corresponding problem set generators): 1) the EO1 spacecraft operations domain, 2) the Rocky-7 Mars rover operations domain, 3) the DATA-CHASER shuttle payload operations domain, and 4) the New Millennium Space Technology Four landed operations domain.

Within each model and corresponding problem set, we generate random problems that include a background set of fixed activities and a number of movable activity groups. The activity groups are placed randomly. The goal is to minimize the number of conflicts in the schedule by performing planning and scheduling operations.

To solve each problem, we use the ASPEN (Automated Scheduling and Planning Environment) system using an "iterative repair" algorithm, which classifies conflicts and attacks them each individually (Rabideau, G., Knight R., Chien S., Fukunaga A., Govindjee A. 1999). Conflicts occur when a plan constraint has been violated; this constraint could be temporal or involve a resource or state timeline. Conflicts are resolved by performing one or more plan modifications such as moving, adding, or deleting activities. The iterative repair algorithm continues until no conflicts remain in the plan, or a timeout has expired.

The planner/scheduler entertains least-conflicting placements when moving activities. In the *control* trials the planner/scheduler does so using by considering singleton activities. We dub this the *atomic* technique. In the *experiment* trials the planner/scheduler is using our aggregation method to compute valid placements for collections of activities. In all cases for each domain, both trials are using the same set of heuristics at all other choice-points (e.g., selection of a conflict or activity group to attempt to repair, where to place within computed valid intervals, etc.). Note that simple (atomic) operations are available in all domains. We now briefly describe each domain including information on the types of activities and resources modeled, what the activity groups are, and how they are interdependent.

EO1 Domain

The EO1 domain models the operations of the New Millennium Earth Observer 1 operations for a two-day horizon. It consists of 14 resources, 10 state variables and total of 38 different activity types. Several activity groups correspond to activities necessary to perform different types of instrument observations and calibrations. The activity groups range in size from 23 to 56 activities, many of which have interactions. For example, taking an image of the earth requires fixing the solar array drive to avoid blurred images. The high-level observation activity group includes both commands to fix the SAD and take the image.

Each EO1 problem instance includes a randomly generated, fixed profile that represents typical weather and instrument pattern. Each problem also includes 8 randomly placed instrument requests for observations and calibrations.

Rocky-7 Domain

The Rocky-7 Mars rover domain models operations of a prototype rover for a typical Martian day. It consists of 14 shared resources, 7 state variables and 25 activity types. Resources and states include cameras (front, rear, mast), mast, shovel, spectrometer, solar array, battery, and RAM. There are four activity groups that correspond to different types of science experiments: imaging a target, digging at a location, collecting a spectrometer reading from target, and taking a panoramic image from a location. Activity group size ranges from 8 to 17 activities. Members in activity groups have positive resource interactions, e.g. opening the aperture for the camera enables subsequently taking a picture. Activity groups also have negative interactions, e.g. several member activities using the onboard buffer. Rover problems are constructed by generating four experiments and randomly generating parameters for the experiments (such as target locations).

New Millennium Space Technology Four Landed Operations Domain

The ST4 domain models the landed operations of a spacecraft designed to land on a comet and return a sample to earth. This model has 6 shared resources, 6 state variables, and 22 activity types. Resources and states include battery level, bus power, communications, orbiter-in-view, drill location, drill state, oven states for a primary and backup oven state, camera state, and RAM. There are two activity groups that correspond to different types of experiments: 1) mining and analyzing a sample, 2) taking a picture. Activity group sizes range from 5 to 10. As in the rover domain, activities interact positively and negatively.

Each ST4 problem instance includes a randomly generated, fixed profile that represents communications visibility to the orbiting spacecraft. Each problem also includes five mining and two picture experiments (each randomly placed.)

DATA-CHASER Domain

The DCAPS domain models operations of a shuttle science payload that flew onboard Space Shuttle Flight STS-85 in August, 1997. It consists of 19 shared resources, 25 state variables, and 70 activity types. Resources and states include shuttle orientation, contamination state, 3 scientific instruments (doors, relays, heaters, etc.), several RAM buffers, tape storage, power (for all instruments/devices), and downlink availability. There is one type of activity group corresponding to one experiment for each of the 3 scientific instruments. This activity group consists of 23 activities. As with the other domains, activities in this activity group interact positively and negatively.

Each DCAPS problem instance includes a randomly generated, fixed profile that represents shuttle orientation and contamination state. The number of randomly placed experiments ranges from 2 to 20 based on the fixed profile for the given problem instance.

For each domain, we run 20 random problems for both the control (atomic) and the experimental techniques. Using the *atomic* technique, the problems are intractable within reasonable time bounds. We postulate that this is because the distance in terms of sub-optimal moves from one local optima to the next is $O(n)$ and the space to be searched is $O(m^n)$ where n is the number of activities in a movable collection and m is the number of possible locations given by a calculation of legal intervals for an individual activity. For example, in the EO1 domain, n ranges from 23 to 56; in the Rover domain, n ranges from 8 to 17.

However, we discover that our board-laying technique fairs somewhat better (see Table 1.)

	EO1	Rocky 7	DCAPS	ST4	total
board laying	149/400	390/400	387/400	243/400	1169/ 1600

Table 1 Number of Problems Successfully Planned

Also, we see that as a function of time, our board laying technique outperforms the atomic technique for each domain (see Figure 6, Figure 7, Figure 8 and Figure 9).

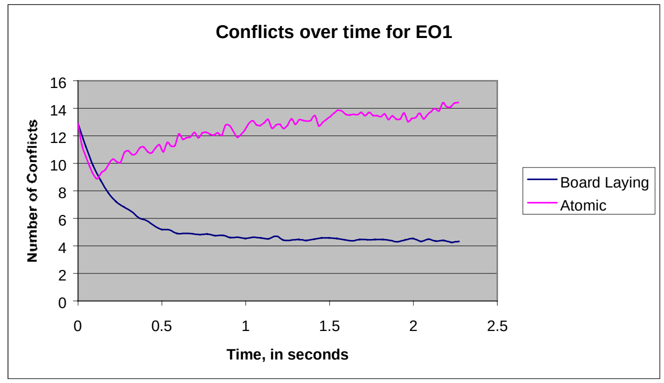

Figure 6 Conflict Reduction for EO1

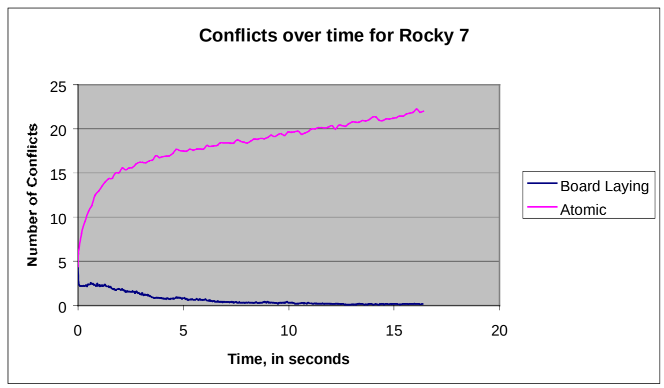

Figure 7 Conflict Reduction for Rocky 7

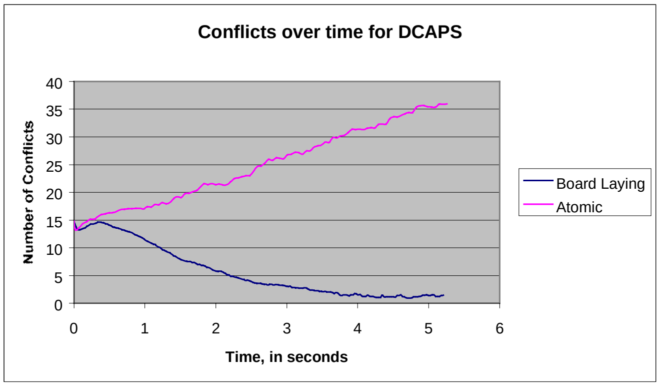

Figure 8 Conflict Reduction for DCAPS

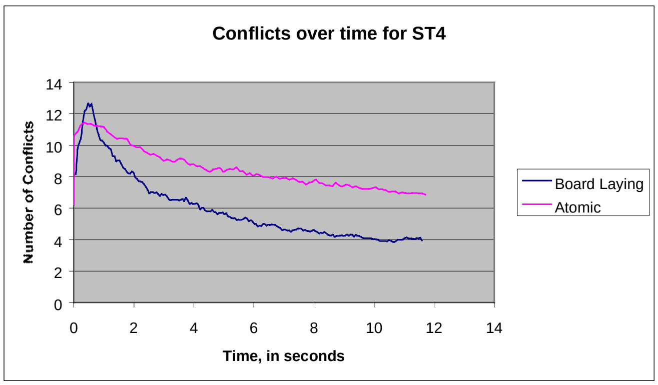

Figure 9 Conflict Reduction for ST4

Related Work

There are a number of related systems that perform both planning and scheduling. IxTeT (Laborie, P., Ghallab, M. 1995) uses least-commitment approach to sharable resources that does not fix timepoints for its resource and state usages.

HSTS (Muscettola, N. 1993) enforces a total order on timepoints affecting common shared states and resources, allowing more temporal flexibility. We believe that our technique is applicable in this case at a greater computational expense (while still being polynomial), and future research should address this issue.

Both IxTeT and HSTS are less committed representations than our grounded time representation and this flexibility incurs a greater computational expense to detect and/or resolve conflicts.

O-PLAN (Drabble, B., and Tate A. 1984) also deals with state and resource constraints. O-PLAN's resource reasoning uses optimistic and pessimistic resource bounds to efficiently guide its resource analysis when times are not yet grounded. Like ASPEN, O-PLAN also allows multiple constraint managers which would enable it to perform general reasoning when times are unconstrained and more efficient reasoning in the case where all timepoints are grounded.

SIPE-2 (Wilkins, D., 1998) handles depletable/non-depletable resource and state constraints as planning variables using constraint posting and reasons at the same level of commitment as IxTeT.

(Cesta, Oddi S., and Smith S. 1998) apply constraint-posting techniques to satisfy multi-capacitated resource problems at the same level of commitment. Depletable/non-depletable resource constraints are easily transformed to multi-capacitated resource constraints. None of these systems generally consider aggregate operations in their search space.

Conclusion

This paper has described the use of board laying techniques to improve the efficiency of planning and scheduling sets of interdependent activities. We show empirically that our board laying search method outperforms the alternative approach of using singleton operations on problems from space exploration domains.

Acknowledgements

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

Cesta, Oddi S., and Smith S. 1998. "Profile-Based Algorithms to Solve Multiple Capacitated Metric Scheduling Problems." *Proc. AIPS98*, pp. 214-223.

Dechter, R., Meiri I., and Pearl J. 1991. "Temporal Constraint Networks," *Artificial Intelligence*, 49, 1991, pp 61-95.

Drabble, B., and Tate A. 1984. "Use of Optimistic and Pessimistic Resource Profiles to Inform Search in an Activity Based Planner," *Proc. AIPS94*.

Estlin, T., Chien, S., and Wang, X. 1997. "An Argument for an Integrated Hierarchical Task Network and Operator-based Approach to Planning," in *Recent Advances in AI Planning*, S. Steel and R. Alami (eds.), Notes in Artificial Intelligence, Springer—Verlag, 1997, pp. 182-194.

Fukunaga, A., Rabideau, G., Chien, S., Yan, D. 1997. "Towards an Application Framework for Automated Planning and Scheduling," *Proc. of the 1997 International Symposium on Artificial Intelligence, Robotics and Automation for Space*, Tokyo, Japan, July 1997.

Kautz, H., and Selman B. 1996. "Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search." *AAAI-96*.

Knight, R., Rabideau G., and Chien S. 2000. "Computing Valid Intervals for Collections of Activities with Shared States and Resources," *Proceedings of the Fifth INternational Conference on Artificial Planning and Scheduling*, 14-17 April 2000, pp. 339-346.

Laborie, P., Ghallab, M. 1995. "Planning with Sharable Resource Constraints," Proc. IJCAI-95, 1643-1649

Muscettola, N. 1993. "HSTS: Integrating Planning and Scheduling." *Intelligent Scheduling*. Morgan Kaufmann, March 1993.

Rabideau, G., Chien, S., Backes, P., Chalfant, G., and Tso, K. 1999. "A Step Towards an Autonomous Planetary Rover," *Space Technology and Applications International Forum*, Albuquerque, NM, February 1999.

Rabideau, G., Knight R., Chien S., Fukunaga A., Govindjee A. 1999, "Iterative Repair Planning for Spacecraft Operations Using the ASPEN System," *iSAIRAS '99*, Noordwijk, The Netherlands, June 1999.

Sherwood, R., Govindjee, A., Yan, D., Rabideau, G., Chien, S., Fukunaga, A. 1998. "Using ASPEN to Automate EO-1 Activity Planning," *Proc. of the 1998 IEEE Aerospace Conference*, Aspen, CO, March 1998.

Tate A., Drabble, B., and Dalton, J. 1996. "O-Plan: a Knowledge-based planner and its application to Logistics," *Advanced Planning Technology, Technological Achievements of the ARPA/RL Planning Initiative*, AAAI Press, 1996, pp. 259-266.

Vidal, V., and Regnier, P., 1999. "Total Order Planning is More Efficient than we Thought." *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, AAAI Press, 1999, pp. 591-596.

Wilkins, D., 1998. "Using the SIPE-2 Planning System: A Manual for Version 4.22." SRI International Artificial Intelligence Center, Menlo Park, CA, November 1998.

Zweben, M., Daun, B., Davis, E., and Deale, M., 1994. "Scheduling and Rescheduling with Iterative Repair," in *Intelligent Scheduling*, Morgan Kaufman, 1994.