AAAI Spring Symposium
Stanford, CA 1989

An Approach to Multiple-Goal Planning with Limited Interactions

Qiang Yang[*] **Dana S. Nau**[†] **James Hendler**[‡]

University of Maryland
College Park, MD 20742

Abstract

The usual approach to multiple-goal planning is to create a single conjoined goal and solve the individual goals as its subgoals. This approach suffers due to the inherent exponential behavior of planning algorithms—solving the conjoined goal takes much longer than solving the individual goals. In this paper, we show that multiple-goal planning problems can be solved more efficiently by generating individual, separate plans for each goal independently and then optimizing the conjunction of these goals, provided that certain restrictions on goal interactions are satisfied. We demonstrate that this approach works well on two different multi-goal planning problems: the problem of finding least-cost multiple goal plans and the problem of finding multiple-goal plans containing the largest possible amount of parallelism.

1. Introduction

Multiple-goal planning systems have long caused problems for artificial intelligence researchers. Systems planning a single task must handle interactions among subgoals of the single plan, but multi-goal systems, solving either a set of potentially interacting goals or a single *conjoined* goal, must handle interactions among both the individual goals and the subgoals of each—a combinatorially explosive proposition [Chapman 1987]. Therefore, it seems unrealistic to expect that general-purpose planners can be developed which can handle a comprehensive set of interactions efficiently. Domain-dependent planners, on the other hand, can often do better at dealing with goal/subgoal interactions in their domain by posing domain-dependent restrictions on the kinds of interactions that are allowed. But the restrictions they use are often either too domain-dependent or too restrictive for the planners to be applicable to other domains.

This paper discusses another approach to multiple-goal planning. We show that generating individual, and separate, plans for each goal independently and then combining the conjunction of these goals is a viable alternative. In particular, we show that although this problem is NP-hard in general, it can be solved more efficiently by imposing restrictions on the kinds of inter-goal interactions involved. Our goal has been to develop restrictions with the following properties:

1. the restrictions are stateable in a clear and precise way (rather than simply referring to general knowledge about the characteristics of a particular domain of application);
2. the resulting classes of planning problems are large enough to be useful and interesting;
3. the classes of problems allowed are "well-behaved" enough that planning may be done with a reasonable degree of efficiency.

As discussed in this paper, we have developed such restrictions for two different multiple-goal planning problems: the problem of finding least-cost multiple-goal plans, and the problem of finding multiple-goal

[*]Computer Science Department. *E-mail address: yang@mimsy.umd.edu.*

[†]Computer Science Department, Institute for Advanced Computer Studies, and Systems Research Center. This work was supported in part by an NSF Presidential Young Investigator award, with matching funds provided by Texas Instruments and General Motors Research Laboratories. *E-mail address: nau@mimsy.umd.edu.*

[‡]Computer Science Department, Institute for Advanced Computer Studies, and Systems Research Center. *E-mail address: hendler@mimsy.umd.edu.*

plans containing the largest possible amount of parallelism. The restrictions, although limiting, are not as severe as the domain-dependent heuristics used by many application-specific planners. With these restrictions, we have developed branch-and-bound algorithms with effective heuristics for solving both of these planning problems.

2. Complexity Analysis

Let $G_1, G_2, \ldots, G_n$ be the goals to be achieved. For each goal G_i, let b_i be its branching factor and d_i its depth [Korf 1987]. If the goals are solved conjunctively, the size of the search space is

$$O((\sum_{i=1}^{n} b_i)^{\sum_{i=1}^{n} d_i}),$$

which is $O((nb)^{nd})$, where $b = \max_{i=1}^{n} b_i$, and $d = \max_{i=1}^{n} d_i$.

On the other hand, if we can solve the goals individually and then combine the solutions to form a global plan, the complexity of the method will be

$$O(nb^d + T),$$

where T is the complexity for identifying and handling interactions between the plans. Moreover, in cases where there are several alternative plans for each goal, T will also include an additional complexity for choosing one plan for each goal in order to satisfy certain optimality criteria. In general, T is still exponential. However, when the goal interactions are limited, we can construct branch-and-bound algorithms with good heuristics for combining and optimizing the plans. As a result, we expect that the total time complexity will be reduced. We demonstrate this claim in Sections 3 and 4.

3. The Least-Cost Multiple Goal Planning Problem

Problem Statement

In this paper, a plan is defined to be a partially ordered set of actions. Actions can have costs, and the cost of a plan is the sum of the costs of the actions. Let G be a goal which is the conjunct of a number of other goals $G_1, G_2, \ldots, G_g$. We assume the plans for the individual goals have already been found, and we look at how to combine them into a global plan.

Depending on what kinds of interactions occur among the actions in the plans, it might or might not be possible for the plans to be combined. In this section, we consider only the following kinds of interactions.

1. An *action-precedence* interaction is an interaction which requires that an action a in some plan P_i must occur before an action b in some other plan P_j.

2. The *identical-action* interaction occurs when an action in one plan must be identical to an action in one of the other plans.

3. Sometimes, two different actions must occur at the same time. We call such interaction a *simultaneous-action interaction*.

4. Let A be a set of actions $\{a_1, a_2, \ldots, a_n\}$. Then there may be a *merged action* $m(A)$ capable of accomplishing the effects of all actions in A. The cost of $m(A)$ could be either higher or lower than the sum of the costs of the other actions—but it is only useful to consider merging the actions in A if this will result in a lower total cost. Thus, although we allow the case where $\text{cost}(m(A)) \geq \sum_{a \in A} \text{cost}(a)$, we can ignore it for the purposes of planning. Thus, we only consider A to be mergeable if $\text{cost}(m(A)) < \sum_{a \in A} \text{cost}(a)$; and in this case we say that an *action-merging interaction* occurs.

   One way in which an action-merging interaction can occur is if the actions in A contain various sub-actions which cancel each other out, in which case the action $m(A)$ would correspond to the set of actions in A with these sub-actions removed. If the cost of each action is the sum of the costs of its sub-actions, then the cost of $m(A)$ is clearly less than the sum of the costs of the actions in A.

   Note that even though a set of actions may be mergeable, it may not always be possible to merge that set of actions in a given plan. For example, suppose a and a' are mergeable, but in the plan P, a must precede b and b must precede a'. Then a and A' cannot be merged in P, because it would require b to precede itself.

Depending on what interactions appear in a given planning problem, it may or may not be possible to combine the plans into a global plan. We call the problem of finding out whether or not a set of plans can be combined into a global plan the ***multiple goal plan existence problem.***

As an added complication, each goal G_i may have several alternate plans capable of achieving it, and thus there may be several different possible identities for the global plan for G. The least costly plan for G_i is not necessarily part of the least costly global plan, because some more costly plan for G_i may be mergeable in a better way with the plans for the other goals.

We define the ***least-cost multiple goal planning problem*** to be the problem of choosing which plan to use for each goal, and which actions to merge in these plans, so as to produce the least costly global plan for G.

The least-cost multiple-goal plan problem occurs in a number of problem domains. As an example, consider a blocks-world problem in which the robot hand has several different grippers, for picking up several different types of blocks. Only one gripper can be mounted on the hand at any given time. Suppose the action pickup(B) consists of mounting the appropriate gripper for block B and then picking up B, and the action putdown(B) consists of putting down B and removing the current gripper. Then the sequence of actions

$$\langle \text{putdown}(A), \text{pickup}(B)\rangle$$

would consist of putting down A, removing the gripper for A, mounting the appropriate gripper for B, and picking up B If A and B were the same kind of block, then these actions could be merged into an action which put down A and picked up B without changing grippers.

Solving the Problems

We have proved that the least-cost multiple-goal plan problem is NP-hard in general, even when we restrict only one plan for each goal. Moreover, when there exists several alternative plans for each goal, the multiple-goal plan existence problem is also NP-hard. One way of handling an NP-hard problem is to simplify it by imposing restrictions on it. The following restrictions will simplify the least-cost multiple goal plan problem:

Restriction 1. Mergeability is an equivalence relation; i.e., it is transitive, reflexive, and symmetric. Thus, we let $A_1, A_2, \ldots, A_k$ be the resulting equivalence classes of actions.

Restriction 2. The global plan Π defines a partial order over the equivalence classes; i.e., if the equivalence classes A_i and A_j are distinct and an action in A_i appears in Π before an action in A_j, then nowhere in Π can an action in A_j appear before an action in A_i. (This does not rule out the possibility of an action in A_i occurring immediately before another action in A_i; in such a case, the two actions can be merged.)

With the above restrictions, we are able to construct a heuristic approach that performs well in practice on these problems. The approach is to formulate the problem as a state-space search and solve it using a best-first branch-and-bound algorithm, as discussed below.

Suppose that we are given the following: (1) for each goal G_i, a set of plans T_i containing one or more plans for G_i, and (2) a list of the interactions among the actions in all of the plans. In the state-space search, the state space is a tree. Each state is a set of plans; it contains one plan for each of the first i goals for some i. The initial state is the empty set (i.e., $i = 0$). If S is a state containing plans for the goals $G_1, G_2, \ldots, G_i$, then an immediate successor of S is any set $S \cup \{P\}$ such that P is a plan for G_{i+1}. A goal state is any state in which plans have been chosen for all of the goals $G_1, G_2, \ldots, G_g$. The cost of a state S is the cost of the plan obtained as the following:

$$\text{cost}(S) = \text{cost}(\text{merge}(\text{combine}(S))),$$

where combine(S) is the result of resolving all the action-precedence, identical and simultaneous action interactions between the plans in S, and merge(P) is the plan after merging all the actions belonging to the same equivalence classes in plan P. A state S is infeasible if combine(S) is no longer partially ordered.

Associated with each state S is a lower bound function L for ordering the members of the list of alternatives being considered. Assuming that merging plans for two different goals always results in a plan at least as expensive as either of the two original plans. If this is true, then clearly $L_0(S) = \text{cost}(S)$ is a lower bound on the cost of any successor of S (this would correspond to using $h \equiv 0$ in the A* search algorithm). However, a better lower bound can be found as follows. Suppose S contains plans for $G_1, \ldots, G_i$. For each $j > i$, let $P^*(S, j)$ be the plan P for G_j which minimizes cost(merge(combine($S \cup \{P\}$))). Let

$$L_1(S) = \max_{j>i} \text{cost}(\text{merge}(\text{combine}(S \cup \{P^*(S,j)\}))).$$

In [Yang, Nau, & Hendler 1988], we show that L_1 is admissible, and show how to compute it efficiently.

4. The Most Parallel Multiple Goal Planning Problem

Problem Statement

A plan is usually associated with *protected conditions*. These are assertions which must remain true during certain time intervals because they are needed either as goals to be achieved or as preconditions for certain actions.

In this section, the only kind of protected condition we consider is one which is *supervised* [Tate 1977], i.e., it is achieved by some action in the plan. For example, if an action a achieves a precondition p of some other action b, then this condition has to be protected from the end of a to the beginning of b. Such a protected condition can be denoted by the triple (p, a, b).

Consider again the problem of planning for multiple goals $G_1, G_2, \ldots, G_n$, where a plan P_i exists for each goal G_i. Suppose some action a in P_i achieves some condition p which is needed by action a', and suppose some action b in P_j achieves some condition s which denies p. Then we say that a *deleted-condition conflict* occurs between G_i and G_j. For simplicity, we disallow the possibility of "white knights" (as defined in [Chapman 1987]), and assume that deleted-condition interactions occur only between pairs of goals. Then there are two possible cases for this conflict, depending on whether s is needed by some other action in P_j.

1. If s is needed by some action b' in P_j, then both p and s are protected conditions (p, a, a') and (s, b, b'), so the conflict is denoted by the pair $((p, a, a'), (s, b, b'))$. In this case, the only way to remove the conflict is to impose a time ordering such that a' occurs before b, or b' occurs before a.

2. If s is not needed by any action in in P_j, then it is not a protected condition, so the conflict can be denoted by the pair $((p, a, a'), (s, b))$. In this case, the way to remove the conflict is to order b before a or after a'.

Although this problem domain is similar to the one described in Section 3, there are several significant differences. On one hand, the deleted-condition conflict is a more general kind of interaction than the action-precedence interaction of Section 3. On the other hand, we do not allow the identical-action, simultaneous-action, and action-merging interactions. Because of these differences, the optimization algorithm developed in Section 3 is not applicable.

For this problem domain, we consider the problem of finding a global plan with the minimum number of ordering constraints between the individual plans. We call such a problem the *most parallel multiple goal plan problem*. For the problem domain of Section 3, this problem can be solved in polynomial by computing combine(S) when there is only one plan for each goal, and is NP-hard if there is more than one plan for each goal. For the current problem domain, the "combine" procedure cannot be used, and the problem is NP-hard regardless of whether or not there is more than one plan for each goal.

As an example. consider the blocks world problem with two goals G_1 = On(A, B) and G_2 = On(E, F). Assume that in the initial situation, the blocks A, B, E and F are all on a table, and C is on A and D is on B. Suppose that block C is too large to fit on any block other than A or E. A plan P_1 for the goal G_1 is move(C, E) and move($D, Table$) in parallel, followed by move(A, B). A plan P_2 for the goal G_2 is move(E, F). A most parallel plan for solving both goals G_1 and G_2 is the action move($D, Table$) and the sequence of actions move(E, F) $\rightarrow$ move(C, E) in parallel, and both are followed by the action move(A, B).

Solving the Problems

Our approach is to formulate the problem as a state-space search problem and solve it using a best-first branch-and-bound algorithm, similar to the one presented in Section 3.

Suppose we are given a plan P_i for each goal G_i, along with a set of protected conditions in each plan. Then the set C of all the deleted-condition conflicts can be found in low order polynomial time in the number of conditions asserted in the plans.

The state space is a tree in which each state is a set of plans, one plan for each goal G_i. The initial state is the original set of plans with no ordering constraint imposed. A goal state is any state for which all of the conflicts in C are removed. If S is any state, then the immediate successors of S are those in which a next chosen conflict from C is removed by imposing partial ordering contraints as discussed previously. State S is infeasible if the newly imposed ordering introduces a cycle in the multigoal plan.

The cost of a state is defined as follows. Let R be the set of pairs (s,t) such that s is an action for a goal G_i, t is an action for another goal G_j, and s is constrained to occur before t. We define cost(S) to be the size of R. It is not hard to see that a goal state with a minimum cost contains a most parallel multiple-goal plan.

We can compute a heuristic $H(S)$ by first updating the set C of remaining conflicts. This is necessary because removing a conflict may also remove some other conflicts in C. Secondly, for each way of removing a conflict c in C, we compute the number of new pairs (s,t) added by imposing an ordering constraint for removing c, such that action s is constrained to precede action t. Let the minimum number of pairs be called N_c^*. Then

$$H(S) = \max_{c \in C} N_c^*,$$

and lower bound function for branch-and-bound search is $F(S) = \text{cost}(S) + H(S)$. This heuristic can be shown to be admissible.

More Than One Plan For Each Goal

It is reasonable to expect that more than one plan may be available for each goal for some planning problems. In cases like this, it is necessary to select one plan for each goal such that the resulting multiple goal plan is the most parallel.

In general, if there are K conflicts existing among a set of plans, and for each conflict there are B ways for resolving it, then the worst case time complexity for finding a conflict-free multi-goal plan is $O(B^K)$. This formula tells us that by minimizing the total number of conflicts to be resolved, we can increase the search efficiency by an exponential amount. In fact, this argument provides us with a good heuristic for choosing among the alternative plans for each goal when solving the most parallel multiple goal problem: We can first search for the set of alternative plans for a set containing the minimum number of conflicts, and then use the branch-and-bound search algorithm discussed in Section 3 for finding the most parallel multiple goal plan.

5. Summary

This paper describes a new approach to multiple-goal planning, based on the idea of generating individual, separate plans for each goal independently and then optimizing the conjunction of these goals. We have shown that with certain restrictions over goal interactions, heuristic search methods can be used for combining and optimizing the multiple goal plans. In domains where such restrictions are satisfied, we expect this approach to be more efficient than solving the goals conjunctively.

Currently we are conducting further experiments with the algorithms discussed in this paper. We are also looking for other planning problems which can be formulated within the framework of our multiple-goal planning approach, as well as ways of lifting some of the restrictions which were imposed on goal interactions.

References

[Chapman 1987] D. Chapman. "Planning for Conjunctive Goals," *Artificial Intelligence* (32). 1987, 333-377.

[Fikes & Nilsson 1971] R. E. Fikes and N. J. Nilsson, "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving." *Artificial Intelligence* (2:3/4). 1971. 189-208.

[Korf 1987] Korf, R.E., "Planning as Search: A Quantitative Approach." *Artificial Intelligence* (33). 1987, 65-88.

[Tate 1977] A. Tate, "Generating Project Networks." *Proc. IJCAI*, 1977, 888-893.

[Nau, Yang, & Hendler 1989] D. S. Nau. Q. Yang and James Hendler, "Planning for Multiple Goals with Limited Interactions," to appear at the Fifth IEEE Conference on Artificial Intelligence Applications, Miami, Florida, March 1989.

[Yang, Nau, & Hendler 1988] Q. Yang, D. S. Nau and James Hendler, "Optimizing Multiple Goal Plans with Limited Interaction," Tech. Report No. UMIACS-TR-88-49, University of Maryland, College Park, July, 1988.