Choosing Objectives in Over-Subscription Planning

David E. Smith
NASA Ames Research Center
Mail stop 269–2
Moffett Field, CA 94035
desmith@arc.nasa.gov

Abstract
Many NASA planning problems are over-subscription problems—that is, there are a large number of possible goals of differing value, and the planning system must choose a subset that can be accomplished within the limited time and resources available. Examples include planning for telescopes like Hubble, SIRTF, and SOFIA; scheduling for the Deep Space Network; and planning science experiments for a Mars rover. Unfortunately, existing planning systems are not designed to deal with problems like this—they expect a well-defined conjunctive goal and terminate in failure unless the entire goal is achieved. In this paper we develop techniques for over-subscription problems that assist a classical planner in choosing which goals to achieve, and the order in which to achieve them. These techniques use plan graph cost-estimation techniques to construct an orienteering problem, which is then used to provide heuristic advice on the goals and goal order that should be considered by a planner.

1. Introduction
Many NASA planning problems are over-subscription problems—that is, there are a large number of possible goals of differing value, and the planning system must choose a subset that can be accomplished within the limited time and resources available. For example, space and airborne telescopes, such as Hubble, SIRTF, and SOFIA, receive many more observation requests than can be accommodated. As a result, only a small subset of the desirable requests can be accomplished during any given planning horizon. For a Mars rover mission, there are many science targets that the planetary geologists would like to visit. However, the rover can only visit a few such targets in any given command cycle because of time and energy limitations, and limitations on the rover's ability to track targets.

Unfortunately, planning systems are generally not designed to deal with over-subscription problems. Most systems expect to be given a well-defined conjunctive goal and attempt to synthesize a plan to achieve the entire goal. They are not able to consider the values of the different goals, or to choose an appropriate subset of the goals that can be accomplished within the limited time and resources available.

In practice, most over-subscription problems have been addressed by using simple "greedy" approaches. For an earth-observing satellite (where slewing is not possible, or slewing times are short) one can create a reasonable observation schedule by considering observations in descending order of their importance or priority. If the observation being considered is still possible, it is added to the schedule; otherwise it is discarded. This approach can work reasonably well for problems in which the cost of achieving an objective does not depend on the order in which the objectives are achieved. Unfortunately, this assumption does not hold for a Mars rover. The reason is that there is a significant cost in moving from one target to the next, and that cost depends on the distance and terrain between the two targets. In other words, the ordering of the targets has a strong influence on the overall cost of visiting those targets. As a result, much more powerful and informed search heuristics are required to help a planner choose the targets to visit and the order in which to visit them.

In this paper, we develop a technique for solving over-subscription planning problems. The technique involves constructing an abstracted version of the planning problem, and then using the resulting solution(s) to provide heuristic advice to the planner on the goals and steps that should be considered, and the order in which they should be considered. The abstracted version of the problem is formed by first estimating the costs of achieving each different objective (goal) using a plan graph. This information is then used to construct an orienteering problem, a variant of a traveling salesman problem. Solutions to the orienteering problem then provide the heuristic information needed for guiding the planner.

In the next section, we discuss plan graph distance estimation techniques, and show why they alone are not adequate for guiding search in over-subscription problems. In Section 3 we introduce the orienteering problem, and show how an orienteering problem coupled with plan graph distance estimation techniques can provide a useful abstraction of a rover planning problem. We then show how the solution to this orienteering problem can provide guidance to the planner search process. Throughout these two sections we limit our attention to a simple rover planning problem, where the mapping between the problem and the orienteering problem is relatively obvious. In Section 4, we generalize the graph construction and solution process so that it
applies to arbitrary planning problems. In Section 5, we present some preliminary experimental results on the rover problem. Finally we discuss related work and some current limitations of the approach.

2. Plan Graph Distance Estimates

A number of recent high-performance planning systems use a plan graph (Blum & Furst, 1997) to compute an estimate of the resources and time required to achieve goals from states encountered in the search process (e.g. Hoffman 2002, 2003; Do & Kambhampati 2003; Edelkamp 2003). This information is used to select among the different alternative search states. To see how this works, consider the simple rover example shown in Figure 1, in which there are three target locations (rocks) with various paths along which the rover can travel. (Assume these paths are precomputed using path planning algorithms.) Both a sample and an image are desired at Loc1, an image is desired at Loc2 and a sample at Loc3.

Figure 2 shows an abbreviated plan graph for the first two levels of this simple problem. Level 1 of the graph shows that it is possible for the rover to reach any of the three target locations in only one step. Level 2 shows that any individual experiment can be achieved after only two steps. Thus, the plan graph provides an optimistic assessment of when actions and propositions are possible. Since actions take time and resources, the graph can be used to compute estimates of the time and resources needed to achieve a goal or objective. In the example in Figure 2, numbers next to the actions indicate the cost of each action. With these numbers, we can use the graph to estimate the cost of achieving each of the objectives at level 2. For example, in order to have a sample at rock 1, we would need to move to rock 1 and then collect the sample, giving a cost of 4+3=7. These cost calculations can be done very rapidly in a plan graph by a simple forward sweep through the graph. During planner search, this kind of heuristic "distance measure" can be used to select between different possible ways of achieving a goal. For example, if the goal is to have the sample at location 3, then it is better to go directly there rather than via either location 1 or location 2. If a direct path is not available, the graph would tell us that it is better to go via location 2 (cost 8) rather than via location 1 (cost 9). It is this idea that provides much of the basis for the search guidance used by the competitive planners in the recent planning competition (Long & Fox, 2003).

For over-subscription problems, we could use the same strategy to estimate the cost of achieving each possible goal from the current state, and then try use this information to select the most appropriate set of goals to achieve. However, we must also take the value of the goals into account. Thus the problem of choosing the set of goals becomes a sort of bin-packing problem in which we are trying to pack the most value into the bins of available resources. For example, suppose that the rover is at location 1, but has only four units of energy available. If we construct a plan graph starting at location 1 and do the cost estimation, the graph will tell us that three goals are possible: Sample1, Image1, and Image2, with costs of 3, 1 and 3 respectively. Solving the (trivial) bin-packing problem, we see that it is possible to achieve either Sample1 & Image1, or Image1 & Image2. If samples are worth more than images, then the first option is better. Otherwise, the second option is better.

While this approach works for this very simple example, it generally does not work well for the rover problem. The reason is that the cost of moving to a target depends heavily on the location of the previous target. Unfortunately, the heuristic distance estimates derived from a planning graph implicitly make the assumption that the goals are independent. For example, if the rover starts at location 0 with eight units of energy, the plan graph tells us that the three objectives are possible: Sample1, Image1, and Image2 with costs 7, 5 and 5 respectively. Based on these costs we would be led to the conclusion that only one experiment can be performed.

---

1. More precisely, these planning systems use this distance information to extract a relaxed plan for the goals, then use this relaxed plan as an estimate of the cost of achieving the goal from the search state.
while in reality it is possible to achieve Sample₁ & Image₁, or Image₂ & Image₃.

The problem is that in the plan graph in Figure 2, the cost estimate for getting to location 1 assumes we are coming from location 0, likewise for locations 2 and 3. However, locations 1 and 2 are close together, so the cost estimate of 4 for getting to location 2 no longer applies if we choose to go to location 1 first. Thus, the plan graph might allow us to pick the best or nearest single goal to accomplish, but it provides little immediate guidance if we want to visit several locations and achieve several goals in succession.

Several researchers have augmented plan graph cost estimation techniques to better account for interaction between actions in the graph. In particular, Hoffman (2002, 2003), Do & Kambhampati (2003), Edelkamp (2003) and Gerevini (2003) extract a relaxed plan from the planning graph and use this plan to estimate cost. However, this technique is primarily aimed at accounting for action duplication in estimating the cost of achieving a well-defined goal. For an over-subscription problem, this technique might improve the estimates for individual goals, but it does not solve the problem of interaction between the goals. The fundamental problem is that the resulting cost estimates assume that the goals are independent, and they are not. Thus, plan graph cost estimation alone does not seem to provide an adequate mechanism for choosing goals and goal ordering in such problems.

3. The Orienteering Problem

To overcome the difficulties mentioned above, we observe that there is a strong similarity between the rover problem and a variant of the traveling salesman problem known as an orienteering problem (Keller, 1989). In an orienteering problem, we are given a set of cities, a prize for each city (possibly zero), and a network of roads between cities. The objective is for a salesman to collect as much prize money as possible given a fixed amount of gas. The orienteering problem has been studied extensively in the operations research literature, and both exact and approximate algorithms have been developed for solving this problem (Keller, 1989). To recast the rover problem as an orienteering problem, the cities become target sites, and the roads are paths between different targets, with costs corresponding to the resources required for the rover to traverse the path. The prizes are the scientific values of the experiments at a given target site. However, since there can be multiple experiments possible at a given site, and there are time and resource costs associated with each experiment, we need to create a separate "city" in the graph for each experiment at a site. We then add directed edges from the site to the experiments at that site, and return edges from the experiments back to the site. The resulting graph for our simple rover example is shown in Figure 3.

We can assign a cost of zero to the return edges from an experiment to a site. However, an edge from a target site to an experiment should be assigned a cost that reflects the time and resources required to perform that experiment. Thus, for the sample at target₁, we label the edge from target₁ to sample₁ with a cost of 3, corresponding to the cost of obtaining sample₁, once we are at target₁. In our simple example, each experiment is only a single step, so it is easy to come up with these numbers. In reality, experiments require many steps, and planning is required to generate these steps. We obtain the numbers for experiment edges in the orienteering problem using plan graph cost estimates, as described in the previous section. In particular, we ignore rover location in the plan graph by assigning a cost of zero to all locations. We can then compute cost estimates for all the objectives in the plan graph. These estimates provide the numbers that we need for each experiment edge in the orienteering problem. We can summarize the approach as follows:

1. Construct a plan graph and use it to estimate the time and resources required to perform the different experiments at each target site. (This can be accomplished for all sites simultaneously by assigning a cost of zero to all locations in the graph.)

2. Construct an orienteering problem, like that shown in Figure 3, using path planning to compute the edge cost for each move between sites, and using the estimates computed in Step 1 for the edge costs between a site node and the science experiments at that site.

3. Solve the orienteering problem and use the resulting solution to guide the planning process.

A solution to this orienteering problem suggests which sites the rover should visit, the order in which to visit those sites, and which experiments should be performed at those sites. This information can then be used as heuristic guidance for a planner.

Searching for Plans

It might appear that solving the orienteering problem provides an exact solution to the rover's planning problem, and that no additional planning would therefore be necessary. While this is true in our very simple example, it is not true in general. There are several reasons for this:

- Some of the steps for an individual experiment may interfere with each other. As a result, the experiment cost estimates obtained from the plan graph may be inaccurate.
Different experiments can share steps or can interfere with each other. For example, obtaining a sample and taking a close up image both involve deploying the arm, so it is possible to share this step if both experiments are performed. This interaction is not modeled by our abstracted orienteering problem, which implicitly assumes that the only interaction between experiments results from the location of the rover.

- There may be time constraints on certain experiments (perhaps due to illumination constraints), or there may be required events (like communication) that must occur at specific times. These constraints are not reflected in the orienteering problem, so the resulting solution may be flawed.

As a result, the orienteering problem is only an abstraction of the real planning problem, and the solution to the orienteering problem may not prove to be a solution to the actual planning problem.

Suppose that we have a solution to the orienteering problem, and use it as heuristic advice to a planner to suggest the goals and the order in which to achieve those goals. When detailed planning is performed, the resulting plan could turn out to be much better or worse than that predicted by the heuristic estimates. In either case, it may be desirable to continue searching for a better plan. One possible approach is to search for another solution to the orienteering problem and try this as heuristic advice. Many of the algorithms developed for solving the orienteering problem (Keller 1989) are based on either local search, or branch and bound, and can therefore be adapted to provide a stream of solutions to the problem. A more ambitious possibility is to update the edge costs for the orienteering problem to reflect the actual values found in planning. One could then solve the orienteering problem again, and use the updated solution to continue guiding the planning process.

4. Generalizing to Multiple Interactions

Thus far, we have assumed that the only strong source of interaction between experiments is the location of the rover. This assumption allowed us to construct an orienteering problem in which the "cities" correspond to locations and experiments. While this is a far better model than that provided using only plan graph cost estimation techniques, it may not provide sufficient guidance for some problems. For example, suppose that a rover instrument takes significant warm up time, but once it is warm, it can be kept warm for additional experiments with little additional cost. In this case, there may be considerable advantage to performing a sequence of experiments at different sites using that instrument. This violates our assumption that "location" is the only rover attribute for which there is strong interaction between experiments. In order to fix this problem, we need to consider instrument-status, along with rover location in our creation of the orienteering problem. Specifically, the cities in the orienteering problem now become location/instrument-status pairs. In effect, we are now solving the orienteering problem on a projection of the state space for the rover — a projection onto the two predicates location and instrument-status.

For our simple rover problem suppose that the instrument is required for the two imaging operations, but not for collecting the samples. The graph would consist of two copies of the orienteering graph from Figure 3, one for the instrument off, and the other for the instrument on. The two graphs would be cross connected by the operations of turning the instrument on or off, as shown in Figure 4. Because the imaging operations require that the instrument be on, these two objectives only appear in the bottom half of the graph, where the instrument is on. However, the sampling operations don't rely on the instrument, so they appear in both the top and bottom halves of the graph.

This brings up an interesting issue: the graph in Figure 4 contains two copies of some of the objectives, Sample1 and Sample2. As a result, it is possible for the solution algorithm to collect the reward for this goal twice, by first visiting it in the upper part of the graph, then transitioning to the lower part (or vice versa). To fix this problem we add mutual exclusion (mutex) edges between all pairs of identical objectives appearing in the graph. We then modify the solution algorithm for the orienteering problem to respect these mutex constraints. This turns out to be fairly simple; the solution algorithm already keeps track of which cities have been visited so that it does not collect rewards twice when returning to a city. All that is necessary is that when a city is visited for the first time, we also add any mutex cities to the set of visited cities. This prevents the collection of any reward when visiting those cities.

Figure 4: Orienteering problem for cross product of location and instrument status.
For our simple rover problem, the structure of the graph in Figure 4 seems fairly obvious. However, if we wish to apply this technique more broadly, we need to be able construct this graph automatically. This turns out to involve several steps.

First, assume that we have the set O of predicates that will form the states in our orienteering problem. What we want to construct is the projection of the problem state space onto that set of predicates. However, this is generally intractable, since it requires that we enumerate the entire state space, project the states onto the desired predicates (get rid of the other predicates), and then combine identical projected states. Instead, we can construct an optimistic approximation of the projected state space by starting with a projection of the initial state, applying all applicable actions to this state, and projecting the resulting states. We repeat this process until no new projected states are found. This algorithm is summarized in Figure 5.

1. Let I = the initial state projected onto the predicates O
2. Let States=<I>, Nodes=<I>, Edges=
3. While States is non-empty
   a. Let s = Pop(States)
   b. For each action a applicable in s:
      Let s' be the projection onto O of the result of applying a to s
      Unless s' is in Nodes, add s' to Nodes.
      Unless s'=s, add the edge <s,s'> to edges

Figure 5: Projected state space construction

For our example, the result of this process is the graph in Figure 4, but without the goal (experiment) nodes and edges.

Next, we need to add the goal (experiment) nodes and edges to the graph. To do this we need to figure out which goal (experiment) nodes connect to which projected states in the orienteering graph. To determine this, we construct a relaxed plan for each goal in the plan graph. However, in doing this construction, we assume that all propositions in the orienteering graph are available in the initial conditions of the plan graph. In other words, in constructing the relaxed plan, we stop the backward search on a proposition if it has cost zero. The resulting relaxed plan will rely on a (possibly empty) set of "initial conditions" or zero-cost propositions belonging to the orienteering graph. This set of propositions corresponds to one or more of the states in the orienteering graph. As a result, we add a copy of the goal to the graph for each such state, and connect it to that state. In our rover example, the relaxed plan for Image 1 relies on the initial conditions Loc1 and Instrument-on. As a result only one copy of the goal is added to the orienteering graph and is connected to the state (Loc1, Instrument-on). In contrast, the relaxed plan for Sample1 relies only on Loc1 so two copies are added, one for the projected state (Loc1, Instrument-off) and one for the projected state (Loc1, Instrument-on). Similar arguments apply to Image2 and Sample2.

An edge from a state node to a goal node is assigned a cost equal to the cost of the relaxed plan for that goal. Thus, for our example, the edges from the state node (Loc1, Instrument-on) to the goals Image1, Sample1, and仪 are the cost of their respective relaxed plans (1 and 3). Finally, we mark any duplicate pairs of goals in the graph as mutex. This algorithm is summarized in Figure 6.

1. Construct a projected state space for the predicates in O
2. Mark all state space propositions as having zero cost in the plan graph
3. For each goal g:
   a. Construct a relaxed plan p for g (halting at zero cost propositions in the plan graph)
   b. Let f be the initial or foundation propositions for p
   c. For each state s in the projected state space consistent with f, add a node g to the graph with reward equal to that of the goal g.
   d. Add an edge from s to g, with cost equal to the cost of the relaxed plan p. Add a return edge from g to s, with cost 0.
   e. Add a mutex edge between all nodes g,

Figure 6: Orienteering graph construction

Identifying Strong Interactions

For the rover problem, it seems fairly clear which attributes should be treated in the orienteering graph and which can be estimated using planning graph cost estimation. However, if we wish to apply these techniques to general planning problems, we need some way of automatically deciding which attributes belong in the orienteering graph. To do this, we can perform a kind of sensitivity analysis on the plan graph to find those attributes that have significant impact on the cost of achieving each goal. For example, in the plan graph shown in Figure 2, consider the relaxed plan for each one of the objectives. We note that all of these plans affect the location of the rover and leave it in a state other than the initial state. We therefore change the location in the initial conditions of the plan graph to see what impact this has on the cost estimates for different objectives. For the goal Sample1, the estimated cost varies from 5 to infinity, depending on the initial location. As a result, location seems like a good candidate for treatment in the orienteering graph. Similarly, if the status of an instrument has a significant impact on the cost of achieving some of the objectives, it too would be a good candidate for the orienteering graph. Using this technique, we can automatically identify those attributes for which the goals strongly interact.

5. Implementation and Experiments

Currently, we have only a partial implementation of the techniques presented in the previous sections. In particular, our implementation does not yet automatically decide which predicates form the basis of the orienteering graph. Further-
more, our automated construction of the orienteering graph currently only handles single predicates. The implementation consists of:

1. Plan graph cost estimation—a simple plan graph is constructed to quiescence for the planning problem.

2. Cost estimation—given a basis set of predicates to be ignored (such as location), costs are computed for the objectives (goals) by setting the costs for ignored predicates to zero and performing the standard forward sweep through the plan graph.

3. Relaxed plan extraction—relaxed plans are extracted from the planning graph for each objective, assuming that all propositions with zero cost are available in the initial conditions.

4. Orienteering problem construction—given a single basis predicate (such as location), an orienteering problem is constructed corresponding to the projection of the state space onto that predicate. This projection is extracted from the propositions and set of transitions present in the plan graph for that predicate.

5. Goal node addition—each objective (goal) is added to the orienteering graph. It is connected to the projected state node in the orienteering graph corresponding to the zero-cost proposition used in the relaxed plan for that goal.

6. Orienteering problem solution—A beam search (using a greedy heuristic lookahead function for evaluation) is used to find solutions to the orienteering problem.

7. Planner guidance—the best solution to the orienteering problem is used to supply the goals to a POCL planner. The goals are fed to the planner one at a time in the order suggested by the solution to the orienteering problem. The planner can link to actions already in the plan structure, but cannot violate existing causal links. Planning terminates when resources are exhausted, or no remaining goals (from the solution to the orienteering problem) can be achieved.

We have performed some preliminary experiments with this system on rover problems involving 10, 25, 50, and 100 rocks randomly distributed in a 50x50 square. Between 1 and 3 experiments were available at each rock with experiment values chosen randomly in the range of 1 to 5. 75% of the n\(^2\) paths between rocks were assumed to be traversable, with costs equal to the distance along the path. For the large problems, we gave the rover sufficient resources to allow it to visit approximately 10% of the rocks. For smaller problems, we tried a range of resource values. We used a beam width of 25 when searching for solutions to the orienteering problem.

In all cases, construction and solution of the orienteering problem is very fast (0.3 seconds for the most difficult problems). Our technique for solving the orienteering problem is approximate, so we are not guaranteed to find the optimal solution. However, by performing experiments with very large beam width, we believe that we are obtaining optimal solutions for smaller problems and solutions within a few percent of optimal for the largest problems. Solution quality tends to drop off with a beam width of less than 15, and a beam width of greater than 50 slows down the solution process significantly. For these problems, a beam width of 25 seems to provide a good compromise between solution speed and solution quality.

We have not yet completed comparisons of the resulting plan quality against plan quality using greedy search strategies. However, our preliminary results indicate that we are typically getting plan quality improvements averaging from 10% to 30% depending on the density of goals. When the field is rich in goals, greedy approaches tend to work reasonably well. But when the goals are widely scattered, the choice of goals and goal order can make a large difference in net reward. Figure 7 shows a typical comparison of plan re-

![Figure 7: Plan quality (reward) for orienteering guidance and greedy guidance of a planner on a moderate sized problem involving 25 rocks in a 50x50 area.](image)

---

October 28, 2003
The difficulty in applying this approach to general over-subscription planning problems is in recognizing which predicates should be part of the orienteering graph, and which can be treated using conventional plan graph cost estimation methods. We presented a technique for making these decisions automatically. It performs a kind of sensitivity analysis in the plan graph to determine how the cost of achieving each objective depends on the achievement of other objectives. If the cost of one or more objectives is highly sensitive to conditions that are changed when achieving other objectives, then those predicates are good candidates for the orienteering graph.

In Section 5, we indicated a number of limitations in our current, very preliminary implementation. We are in the process of removing these limitations and expect to have a more complete working system in the next several months. There are, however, some deeper issues that we have not yet addressed. As we mentioned in Section 3, time constraints on objectives, and on required activities such as communications, are not considered in the orienteering graph. Although there has been little work on solving orienteering problems with time constraints, it seems likely that some of the algorithms could be adapted to deal with them. This could further improve the accuracy of the approximate solutions, and thereby produce better search guidance for a planner. However, there is likely to be a cost to this increased accuracy, and it remains to be seen whether this additional accuracy will pay off.

8. Acknowledgments

Thanks to Maria Fox, Nicolas Meuleau and Sailesh Ramakrishnan for discussions on this topic. This research was supported by NASA Ames Research Center and the NASA Intelligent Systems program.

9. References


October 28, 2003


