Submission to IWPSS-2006

Mission Operations Planning with Preferences: An Empirical Study

John L. Bresina, Lina Khatib*, and Conor McGann*

NASA Ames Research Center
M/S 269-2, Moffett Field, CA 94043
John.L.Bresina@nasa.gov

Abstract

This paper presents an empirical study of some non-exhaustive approaches to optimizing preferences *within* the context of constraint-based, mixed-initiative planning for mission operations. This work is motivated by the experience of deploying and operating the MAPGEN (Mixed-initiative Activity Plan GENerator) system for the Mars Exploration Rover Mission. Responsiveness to the user is one of the important requirements for MAPGEN; hence, the additional computation time needed to optimize preferences must be kept within reasonable bounds. This was the primary motivation for studying non-exhaustive optimization approaches.

The specific goals of the empirical study are to assess the impact on solution quality of two greedy heuristics used in MAPGEN and to assess the improvement gained by applying a linear programming optimization technique to the final solution.

Introduction

This paper presents an empirical study of some non-exhaustive approaches to optimizing preferences within the context of constraint-based, mixed-initiative planning for mission operations. Our motivation derived from the problem of activity planning for the Mars Exploration Rover (MER) mission and the system used to accomplish this task: MAPGEN, Mixed-initiative Activity Plan GENerator (Bresina, et al., 2005a). Responsiveness to the user is one of the important requirements for MAPGEN; hence, the additional computation time needed to optimize preferences must be kept within reasonable bounds. This was the primary motivation for studying non-exhaustive optimization approaches. A secondary concern was to incorporate preference optimization into MAPGEN without major changes to the planner's search algorithm.

The MAPGEN system represents a successful mission infusion of mixed-initiative planning technology. MAPGEN was deployed as a mission-critical component of the ground operations system for the Mars Exploration Rover (MER) mission. Each day, the Tactical Activity Planner (TAP) employs MAPGEN to collaboratively plan the activities of the Spirit and Opportunity rovers, with the objective of achieving as much science as possible while ensuring rover safety and keeping within the limitations of the rover's resources (e.g., power).

The MER mission has been operating with great success for over two years, and MAPGEN continues to be employed for activity plan generation for the Spirit and Opportunity rovers. During the multi-year deployment effort and subsequent mission operations experience, we have learned valuable lessons regarding application of mixed-initiative planning technology to mission operations (Bresina, et al., 2005b). These lessons have stimulated new research in mixed-initiative planning with preferences.

The MER scientists express their intent to the MAPGEN system through the requested activities, the associated priorities, and science constraints. By enforcing the specified science constraints, MAPGEN ensured that the data collected satisfied the science intent. However, in addition to these hard constraints, the scientists often have temporal preferences in mind, which could yield higher quality data. Such temporal preferences cannot be formally encoded in MAPGEN. Some of these preferences are verbally communicated to the TAPs, and if they have time, they try to satisfy them by fine-tuning the plan. In addition, there are other more global preferences related to solution quality that were not formally encoded and were left up to the TAPs to satisfy.

We have extended our research version of MAPGEN by enabling the system to enter temporal preferences and are exploring alternative techniques for optimizing the satisfaction of (possibly competing) preferences. In this paper, we focus on non-exhaustive approaches that are more efficient and easier to integrate into MAPGEN, and we present the results from an empirical study aimed at evaluating these approaches. Specifically, the goals of the empirical study are the following: (*i*) assess the impact on solution quality of a greedy priority-based heuristic used in MAPGEN, (*ii*) assess the additional impact of a greedy preference-based heuristic used in MAPGEN, and (*iii*) assess the improvement gained by applying a linear programming technique to the final solution. In order to make it easier to perform a controlled empirical study, we employed analogs of MAPGEN and the MER planning problems.

In the next section, we present background material on aspects of MAPGEN that are relevant to the empirical study, the representation and types of temporal preferences, and our linear programming optimization technique. In the subsequent sections, we describe the design of the empirical study and discuss the results. We close with some concluding remarks.

---

*Affiliated with QSS Group Inc.

Background

The core of the plan representation and reasoning capabilities in MAPGEN is a constraint-based planning framework called EUROPA (Extendable Uniform Remote Operations Planning Architecture), developed at NASA Ames Research Center (Jónsson, et al., 1999; Frank and Jónsson, 2003).

In constraint-based planning, domain rules are specified in terms of activity/state patterns and constraint schemas. A given constraint schema is applied to any instance matching the associated pattern. Search methods and other techniques for manipulating partial plans then build on this framework.

The science constraints are relations between specific activities in a planning problem instance. The scientists use two types of science constraints: temporal bounds and temporal ordering relations. The temporal bounds are typically constraints on when an activity can start due to, for example, lighting conditions or temperature. The typical ordering relations are constraints between the end of one activity and the start of another. For example, a hazcam documentation image of an arm placement must be taken at least two minutes after the arm is placed (to ensure vibrations have subsided) and before it is moved again.

Consistency of the developing plan is maintained using an underlying simple temporal constraint network, or STN [4]. One advantage of STNs is that rather than doing simple consistency checking, they work by eliminating inconsistent values from variable domains. Specifically, they maintain *arc-consistency*, which for STNs is equivalent to full consistency. In effect, they maintain a family of related solutions, called a *flexible* solution, rather than just a single *grounded* solution. A flexible solution provides flexibility because it can often merely be refined, i.e., further restricted, in response to additional constraints instead of requiring search for a new solution.

Minimum Perturbation Heuristic

Although MAPGEN constructs flexible plans, the plan that is displayed to the user is a grounded solution; i.e., a specific consistent instantiation of the underlying flexible plan. This is selected to be as close as possible to an internally maintained *reference schedule*. More importantly, the reference schedule is used to support a *minimum perturbation* approach, where planner-initiated changes to the previous plan are minimized. Users tend to expect that small extensions to a plan will cause only minor plan modifications and dislike it when they cause drastic global modifications. The *minimal perturbation heuristic* biases the ordering decisions such that the activities remain as close to their reference times as possible.

The reference schedule is initially based on the science constraints and the initial start times of the activities, which are set by the scientists. This initial reference is computed by first solving a relaxed version of the

planning problem composed of only the science constraints; the solution produced is a flexible plan. The reference schedule is determined by grounding this flexible plan, by the following *solution grounding algorithm*, to be close to the initial activity start times.

For each timepoint x with reference position t do the following:
(*i*) If t is within the STN bounds for x,
   then add a grounding constraint that sets x to t.
   Else if t is less than the lower bound (lb) for x,
   then add a grounding constraint that sets x to lb.
   Else if t is greater than the upper bound (ub) for x,
   then add a grounding constraint that sets x to ub.
(*ii*) Propagate the effect of the new constraint.

The scientists can bias the initial reference schedule to reflect their preferences. One option is to bias the placement of activities to be when solar power is at a maximum by setting all start times to the time of peak power. During the planning process, the reference schedule is continually updated to reflect the evolving plan.

Priority Heuristic

A key factor in the design of MAPGEN planning methods was that the set of science observation requests oversubscribes available rover resources and, thus, each activity has a given priority that had to be taken into account. The assigned priority is based on the science team's judgment of relative importance; the MER scientists used five different priority levels.

The priorities are the dominant factor in assessing solution quality. The priorities are treated lexicographically; that is, getting one activity at a given priority level into the plan is worth more than any number of activities of lower priority.

During planning, the priorities are used to determine the order in which activities were planned. Furthermore, if a planning request cannot be completed, MAPGEN can reject lower-priority activities to make room for higher-priority activities in the plan.

Preferences

We have extended the research version of the Constraint Editor to allow specifying temporal preferences on an activity's start or end time, as well as on distances between start/end time points of two activities (see Figure 1). In particular, we have enhanced the Constraint Editor tool to allow specification of a *sweet spot* in addition to a base constraint. The sweet spot is an interval of maximum preference and outside the interval, the preference drops linearly from its maximum value. Note that a sweet spot in a science preference can be a single point to indicate, for example, the preference to start as early or late as possible. This preference function
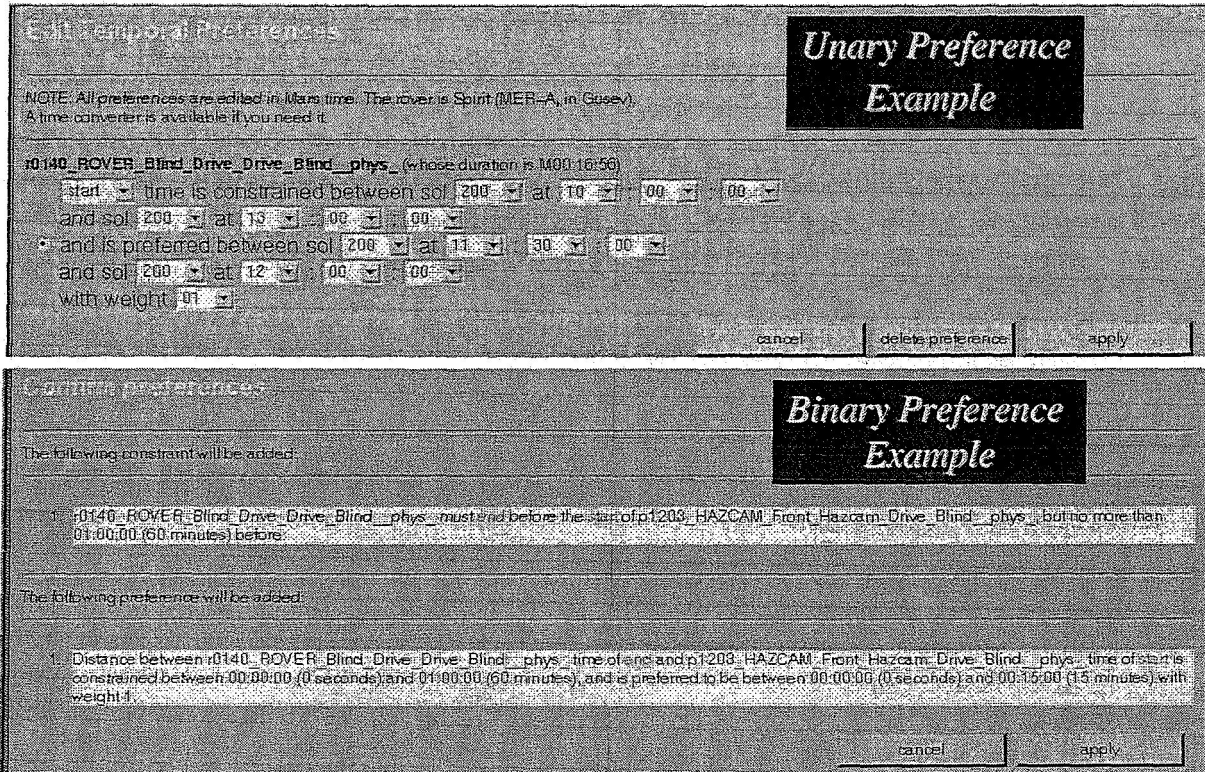
Figure 1: Editing science preferences in CE.

Science Preference

Preferred interval *(sweet spot)*

Preference value — 1

— 0

Constrained interval *(scientifically valid)*

Reference Preference

Reference Time

Preference value — 1

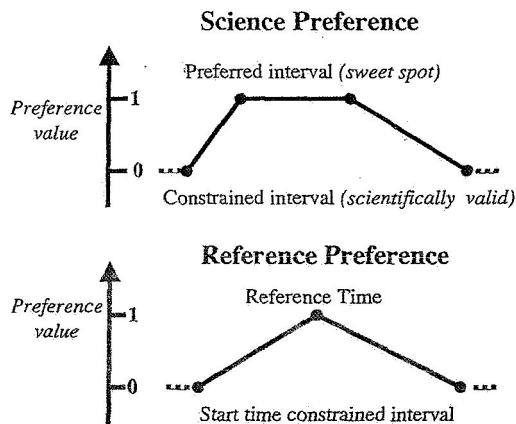— 0

Start time constrained interval

Figure 2: Preference functions

form corresponds to the temporal preferences most frequently expressed by the MER scientists.

The preferences represented by the initial reference schedule can also be represented in this same way. In this case, the temporal preference is on an activities start time and the sweet spot is a single timepoint determined by the reference schedule. In order to ensure that an early and a late deviation (from the reference time) of the same magnitude have equal preference, the zero crossings of the preference function must be equidistant from the reference time. To accomplish this, one of the zero crossings may lie outside the base constraint. We refer to these temporal preferences as *reference preferences*.

Figure 2 illustrates the form of a typical science preference and typical reference preference. In this figure, the preferences have a maximal value of one; however, each preference is associated with a weighting factor, which determines the relative impact of the preference during optimization.

Utilitarian Optimization

To effectively solve constraint problems with preferences, it is necessary to be able to order the space of assignments to times based on some notion of global preference and to have a mechanism to guide the search for solutions that are globally preferred. Such a framework arises as a simple generalization of the Simple Temporal Problem (STP) (Dechter, Meiri, and Pearl, 1991), in which temporal constraints are associated with a local preference function that maps admissible times into values; the result is called *Simple Temporal Problem with Preferences (STPP)* (Khatib, et al., 2001). Globally optimal solutions to STPPs emerge as a result of well-defined operations that compose and order partial

solutions. Different concepts of composition and comparison result in different characterizations of global optimality. One natural criterion is utilitarian, where the global value of a solution is the sum of the local values.

It has been shown in (P. Morris, et al., 2004) that determining the set of all utilitarian optimal solutions as an STP is tractable where all the preference functions are convex and piecewise linear, which is the case for our preference functions. The paper shows that this utilitarian optimization problem can be reduced to a Linear Programming Problem (LPP), which is known to be solvable in polynomial time by Karmarkar's Algorithm (Corman, Leiserson, and Rivest, 1990). Furthermore, the paper shows that constructing the STP representing all optimal solutions (i.e., the optimal flexible plan) can be accomplished by adding constraints to the STP, and that the constraints to add can be determined by solving the *dual* of the original LPP.

In our research version of MAPGEN, we have incorporated an optimization technique based on this approach. Given a flexible plan produced by MAPGEN, this new facility further restricts the plan to one containing only utilitarian-optimal solutions with respect to a set of given temporal preferences.

Empirical Study Design

The underlying question that initially motivated this empirical study is: "Can we incorporate preference optimization into MAPGEN without undue impact on the responsiveness of the system to the user?" A secondary concern was to incorporate this new capability without major changes to the planner's search algorithm.

We were interested in trying to incorporate the linear programming based utilitarian optimization into MAPGEN and designed a couple of ways this technique could be employed. One use is to apply the optimization, as a post-process, to the family of solutions represented by a flexible MAPGEN plan in order to display the most-preferred grounded solution to the user. The technique can also be employed in a pre-processing step to compute a different type of reference schedule — one that represents a globally optimal solution to the relaxed planning problem with temporal preferences (science and reference). Hence, the minimal-perturbation method could be employed to bias the planning decisions so as to stay close to this "ideal" reference schedule.

The search algorithm in MAPGEN already includes aspects that could be employed to support the optimization of preferences; namely, the activity priority heuristic, as well as the reference schedule mechanism and the associated minimal perturbation heuristic. Hence, one question to be answered is what is the impact of each of these two greedy heuristics on the quality of solutions generated, when the quality criteria included preferences in addition to the priorities of the activities that made it into the plan.

In order to achieve the goals of the empirical study, we needed to be able to vary the configuration of the problem solver. The MAPGEN system was not designed to be configurable in this way; hence, it was not practical to use the system directly. Thus, we built a configurable problem solver that is an analog to the planner in MAPGEN. We also wanted to be able to control the composition of the suite of problem instances; this was easier to accomplish with an analog to the MER domain model and a problem generator for this analog model.

Problem Suite

The domain model for the study has one activity type, called *TakeSample*. The planning horizon for all problems is between 10:00 and 16:00 (local time), and all activities in the plan must occur within this time span. A problem instance consists of the following aspects:

- A set of TakeSample instantiations, each with a set duration and priority.
- A set of precedence ordering constraints between pairs of TakeSample activities.
- A set of temporal bound constraints on the start time of the TakeSample activities.
- A set of science temporal preference functions (i.e., "sweet spots") w.r.t. the start times.
- A set of reference temporal preferences w.r.t. the start times.

In the problem suite, all instances have forty TakeSample activities, and all the initial start times are set to noon. Thus, the computed reference preferences will be biased to schedule activities near the solar power peak time. Note that the reference preferences will vary across problem instances since the set of temporal constraints (bounds and orderings) will vary. Each activity is assigned a priority randomly chosen from the set $\{1, 2, 3, 4, 5\}$, where 5 is the highest priority.

There are three control parameters for the generator: the number of precedence ordering constraints, the maximum activity duration, and the maximum percentage of the sweet spot.

For each problem, the generator assigns each activity a duration, randomly chosen between one minute and the specified maximum value. The ordering constraints are generated by randomly choosing the specified number of pairs of unordered activities and imposing a precedence constraint between them. The transitive closure of the ordering constraints is maintained during this process, so that two activities are considered unordered only if no explicit precedence or transitive ordering exists between them.

The set of constraints must be consistent; to ensure this, the generation of temporal bounds on start times is based on the generated ordering constraints, as follows. The activities are first temporarily assigned random start times, restricted such that they obey the precedence orderings. Second, for each activity a lower bound is

randomly chosen within the interval [10:00, S], where S is *the activities assigned start time*. Likewise, an upper bound is randomly chosen within the interval [S, 16:00]. Note that it is possible for a bound constraint to equal the planning horizon, and it is possible for an activity's start time to be restricted to a single timepoint.

The science preferences are generated by first randomly choosing the sweet spot's length between zero and the specified maximum. Second, the lower bound is randomly chosen such that both sweet spot bounds are within the bounds of the associated hard constraint. Note that the sweet spot may be a single point.

The complete problem suite is generated from the following control parameter ranges:

- Constraint Count (CC) – The number of ordering constraints to add between activities. The values are {20, 40, 60, 80, 100}
- Duration Bound (DB) – The maximum activity duration allowed. The values are {500, 1500, 2500, 3500, 4500}
- Sweet Spot Percent (SS) – The maximum percentage of the *start time bounds* that can be used for the sweet spot in expressing a temporal preference. The values are {25, 75}.

This yields fifty control parameter combinations, or problem types. For each problem type, ten distinct problem instances are generated. The *ten scores are* averaged to obtain a score for the problem type.

The primary problem characteristics of interest are the degree of oversubscription, the number of alternate solutions, and the variance of quality in the solution space. If it is easy to fit all the activities into a plan, then the priority heuristic will not have much, if any, impact on solution quality. If there are very few solutions or very little variance in solution quality, then optimizing techniques will not have much affect.

Given that the problems all have the same number of activities, we can indirectly affect these characteristics with the three control parameters of the generator. Increasing the number of ordering constraints will tend to reduce the number of solutions, and increasing the maximum activity duration will tend to increase the degree of oversubscription. The affect of the control parameters on the quality variance in the solution space is more difficult to predict because they all interact.

Solution Quality Function

We present the priority and preference solution quality factors separately so that we can better illustrate the impact of the different problem solver configurations. Both factors are defined in terms of a number of individual attributes and evaluated with respect to a *grounded* solution.

Each quality attribute is assigned a real value between zero and one that indicates the degree to which it is satisfied, where one indicates full satisfaction. Each *attribute is weighted based on its relative importance.* Each of the two factors is a summation of all the associated weighted attribute values.

As was the case in the MER mission, the activity priority levels are treated lexicographically. There is a *priority attribute for each activity.* The attribute value is one if it is in the plan and zero if not. The attribute weight is ten raised to the power of the associated activity's priority.

For the purpose of this experiment, we assume that all science preferences have equal importance and that all reference preferences have equal importance. Furthermore, we want each of the two preferences classes to have equal impact on the overall score. We weight each science preference by one divided by the number of science preferences; similarly, we weight each reference preference by one divided by the number of reference preferences.

Solver Configurations

All the configurations are based on a EUROPA2 Solver, which uses chronological backtracking. As a *baseline* configuration, we are using a Solver without any heuristic bias. The planning order of the activities is randomly selected, and which activities get into the plan is random; i.e., it is not based on the activity priorities. The temporal preferences are ignored and the ordering decisions are made arbitrarily.

The second configuration is a *priority-only* solver; it is a customization of the baseline solver that includes an analog of MAPGEN's priority heuristic. This solver determines planning order based on activity priority. As in the baseline solver, ordering decisions are still made arbitrarily.

The third configuration is a *priority-plus-preference* solver, built from the priority-only solver by adding a greedy technique for satisfying temporal preferences. This solver uses an analog of MAPGEN's minimal perturbation heuristic biased towards an "ideal" reference schedule; we refer to this analog as the *preference heuristic.* As described at the beginning of this section, the ideal reference schedule is produced via our LP optimization technique and it represents a globally optimal solution to the relaxed planning problem with temporal preferences. Hence, in this solver, the planning order is determined by priorities, and the ordering decisions are biased so that activities stay as close to their ideal reference time as the hard constraints allow.

Evaluation Methodology

The primary aims of the empirical study are to evaluate the impact of the priority heuristic, the

preference heuristic, and the LP optimization. We first measure the impact of using the priority heuristic to determine planning order by comparing the performance of the baseline solver with that of the priority-only solver. We then measure the additional impact of using the preference heuristic to bias ordering decisions during search by comparing the performance of the priority-only solver with that of the priority-plus-preference solver.
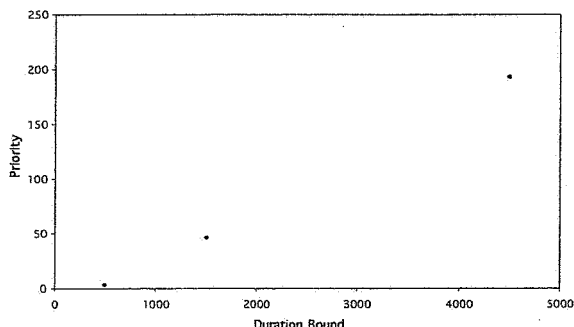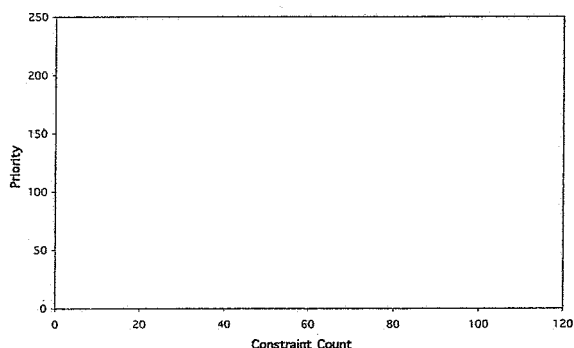

Figure 3: Priority Improvement (1)

Figure 4: Priority Improvement (2)

Note that each of these three solvers constructs a flexible plan; hence, we cannot directly apply the solution quality function to obtain a performance measure. Each flexible plan represents a set of ground solutions of (most likely) varied quality. The quality measure that we've chosen is the expected quality of the flexible plan's execution, and we estimate this measure via sampling. Each sample yields a randomly chosen execution trace as follows. At each execution step, first, the set of activities that is eligible to execute next is determined from the flexible plan's underlying constraint network. One of these activities is chosen randomly and its execution is simulated by advancing time. The impact of grounding the activity's start time is also propagated through the constraint network. Each sampled execution trace

corresponds to a grounded plan and can, thus, be scored with our solution quality functions. The mean of the resultant set of scores is an estimate of our expected quality measure.

In addition to using the mean quality scores to compare two solvers, we use the mean scores to measure the added impact of LP optimization for each of these three solvers. Applying the LP optimization technique to the flexible plan output of a solver determines the optimal quality score achievable with the flexible plan. We compute the improvement, expressed as a percentage, obtained by this LP optimization post-process as one hundred times the difference between the optimum and the mean, divided by the mean.

Another important aspect of problem solving performance, besides solution quality, is computational cost. Thus, as part of this empirical study, we compare the different configurations with respect to computation time.
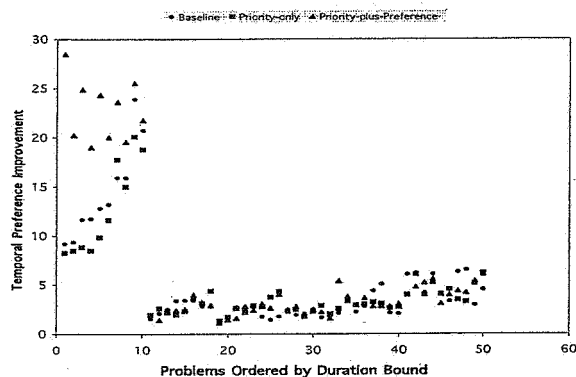

Figure 5: Post-process LP Optimization Impact

Empirical Results

Figure 3 shows the improvement gained by employing the priority heuristic. For a given problem, the improvement is computed as one hundred times the difference between the priority-only solver's priority score and the baseline solver's priority score, divided by the baseline solver's priority score. For a given value of the duration bound control parameter, there are ten problem types and, hence, one hundred problem instances. In Figure 3, the average improvement over the one hundred problem instances is plotted for each of the five values of the duration bound control parameter. The results clearly show the strong increase in performance of the priority heuristic as the problem becomes increasingly over-subscribed.

Figure 4 illustrates the same improvement computation in relation to the constraint count control parameter.

Interestingly, the results indicate that the priority heuristic performs well across all constraint counts.

Figure 5 illustrates the impact of applying the post-process LP optimization to the flexible plan generated by each solver configuration. The percentage improvement (defined in the previous section) is plotted for each of the fifty problem types; hence, it is an average over the ten instances of that type. The x-axis in Figure 5 orders the problem types in terms of increasing values of the duration bound parameter.

The results are striking in terms of the strong negative correlation between LP improvement and duration bounds. For problems with a duration bound control value of 500, LP provides a markedly greater improvement than for higher values, where it appears to offer little value. Moreover, the improvement using LP post-processing is greatest for configurations using the preference heuristic during search. These data suggest that as activity durations grow larger, the available slots in which to place an activity diminish in a way that greatly limits the capabilities of the preference heuristic. The data further indicate that the preference heuristic compliments the post-process LP optimization. An explanation for this effect is that, where such a heuristic is effective, the flexible solution generated is more likely to contain very good solutions (w.r.t. preference score); thus, LP has a better optimal solution to find.

Figure 6 illustrates the improvement obtained by employing the preference heuristic; that is, the percent improvement gained by the priority-plus-preference solver as compared to the priority-only solver. The improvement in preference score is plotted for both the mean value of the flexible plans and the optimal value obtained by the post-process LP optimization. The problem types along the x-axis are ordered by duration bound and, within each duration bound, the problem types
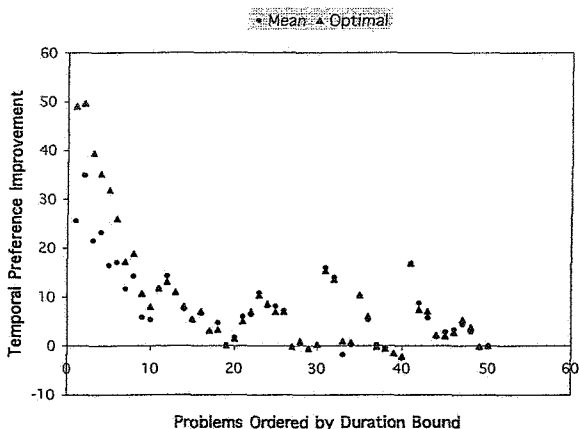

Figure 6: Temporal Preference Improvement

are ordered by constraint count. The data strongly supports the expectation that the impact of the preference heuristic decreases as the problem becomes more highly constrained. The effects of increased activity duration (i.e., greater over-subscription) are dominant. Secondly, as the constraint count increases, the improvement obtained by the preference heuristic further declines.

Table 1 shows aggregate results on the costs and benefits of LP optimization and the preference heuristic. Recall that the priority-plus-preference solver requires an additional run of LP to find an initial optimum solution to the relaxed problem in order to seed the heuristic (i.e., initialize the reference schedule). We do not show the costs of the priority heuristic compared to the baseline since its overhead is negligible.

	MEAN	STD DEV
LP OVERHEAD	40.5506	29.8014
TP OVERHEAD	15.2121	62.9793
LP ROI	14.2524	9.71664
TP ROI	5.44118	26.0244

Table 1: LP Optimization Costs and Benefits

The first row of the table shows the mean and standard deviation of the overhead for the post-process LP optimization. The overhead, expressed as a percentage, for a particular run of one of the solvers is computed as the one hundred times the CPU time used by the post-process LP optimization, divided by the CPU time used to generate the (flexible) solution. The data in the table is aggregated from all three solvers applied to the five hundred problem instances. The data shows a large mean overhead-percentage as well as a large standard deviation.

A similar method is used to compute the overhead of employing the preference heuristic. However, in this case the calculation for each sample compares data from the priority-only solver and the priority-plus-preference solver only. Here, the mean overhead-percentage is low but the standard deviation is very high by comparison.

Rows three and four of the table correlate data on the improvement of each technique compared to the overhead cost. We refer to this as the *Return On Investment (ROI)*. The ROI is calculated in each case as one hundred times the improvement, divided by the overhead. Given that we saw such a strong drop-off in performance of these techniques at duration bounds of 1500 or more, and given the significant mean and variance in the overhead, it is not surprising that the mean ROI for each is low and the variance is high.

Concluding Remarks

There are two main findings of this work. First, we find that the priority heuristic is increasingly effective as the problem becomes more over-subscribed. In a mission where resources are over-subscribed and activity preferences have a lexicographic ordering, this heuristic is very relevant. Second, we find the value of using the post-process LP optimization or of using the preference heuristic during search is more dependent on the problem-solving context. The costs of LP were high by comparison to solving the single planning problem, and the benefits diminished quickly as the problem became more highly constrained. However, in many problem-solving contexts the improvement quality gained is well worth the wait.

There are a number of additional interesting observations to make. The baseline solver did not employ heuristics to improve search efficiency. We did not find any degradation in search efficiency across solver configurations. When heuristics can be effectively used to improve search efficiency, there may be a conflict with the methods evaluated to improve solution quality; thus leading to a greater overhead to achieve quality improvements. The rapid degradation of performance of the preference heuristic based on activity duration was surprising. We have not studied cases of highly over-subscribed problems with small activity durations. This might yield better performance from this technique and we plan to evaluate this question in further experiments. Finally, our work focused on non-exhaustive techniques. We plan to explore uses of branch-and-bound algorithms to find complete optimal solutions.

References

J. Bresina, A. Jónsson, P. Morris, and K. Rajan, "Activity Planning for the Mars Exploration Rovers", *Fourteenth International Conference on Automated Planning and Scheduling*, Monterey, 2005, pp. 40-49.

J. Bresina, A. Jónsson, P. Morris, K. Rajan. "Mixed-Initiative Planning in MAPGEN: Capabilities and Shortcomings." *Workshop on Mixed-Initiative Planning and Scheduling*, ICAPS05, 2005.

T.H. Cormen, C.E. Lieserson, and R.L. Rivest, *Introduction to Algorithms*, MIT Press/McGraw-Hill, 1990.

R. Dechter, I. Meiri, and J. Pearl, "Temporal constraint networks", *Artificial Intelligence*, 49, May 1991, pp. 61–95.

J. Frank and A. Jónsson, "Constraint-Based Interval and Attribute Planning." *Journal of Constraints Special Issue on Constraints and Planning*, 2003.

A. K. Jónsson, P. H. Morris, N. Muscettola, and K. Rajan, "Next generation Remote Agent planner", *Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS99)*, 1999.

L. Khatib, P. Morris, R. Morris, and F. Rossi. "Temporal reasoning about preferences." *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, Seattle (WA, USA), 2001. Morgan Kaufmann, San Francisco, CA.

L. Khatib, P. Morris, R. Morris, and B. Venable, "Tractable pareto optimization of temporal preferences." *Eighteenth International Joint Conference on AI*, Acapulco, Mexico, 2003.

P. Maldague, A. Ko, D. Page, and T. Starbird, "APGEN: A multi-mission semi-automated planning tool." *First International NASA Workshop on Planning and Scheduling*, Oxnard, CA, 1998.

P. Morris, R. Morris, L. Khatib, S. Ramakrishnan, and A. Bachmann, "Strategies for Global Optimization of Temporal Preferences", *Tenth International Conference on Principles and Practices of Constraint Programming (CP-2004)*, Toronto, Canada, 2004.