ARTICLE

MEXAR: Integrated AI Technologies to Support MARS EXPRESS Mission Planning

Authors: **G. Cortellessa, N. Policella, A. Cesta, and A. Oddi**

Introduction

Space exploration missions require coordination of a significant amount of activities. State of the art intelligent planning and scheduling (P&S) technology could potentially be of great help in supporting such a coordination. This work aim at showing an example of this technology in a support system for a specific mission scheduling problem related to the ESA program called MARS-EXPRESS [3].

MARS-EXPRESS is a space probe that will be launched during 2003 and after six months will be orbiting around Mars for the following two years and more. During the operational phase around Mars a team of people, the Mission Planners, will be responsible for the on board operations of MARS-EXPRESS. They receive input from different teams of scientists and cooperate with different specialists for various specific tasks (e.g., Flight Dynamics (FD) experts). Any single operation of a payload, named POR (Payload Operation Request), is decided well in advance through a negotiation phase among the different actors involved in the process (e.g., scientists, mission planners, FD experts).

The result of our study is a system called MEXAR that addresses the memory dumping problem in MARS-EXPRESS. The specific problem that is addressed is defined as MEX-MDP and is described in detail in [7]. MEXAR is an interactive support system that may help mission planners in deciding policies for downlinking data to Earth during the temporal visibility windows. The tool uses constraint-based techniques for representing the basic problem to be solved, namely the segmentation of on-board memory in data packets during the visibility toward Earth.

The paragraph below introduces the two solving algorithms which have been developed: a greedy heuristic and a local search procedure [7, contains a complete explanation of these algorithms]. The following section describes an important aspect of this work, the interactive functionalities developed to support the user in his work [1, for more details].

The Packet Sequencing Algorithm

Scheduling problems such as MEX-MDP can be seen as a special types of Constraint Satisfaction Problems (CSP) [6]. An instance of a CSP involves a set of variables $X = \{X_1, X_2, \ldots, X_n\}$, a domain D_i for each variable and a set of constraints $C = \{C_1, C_2, \ldots, C_q\}$ s.t. $C_i \subseteq D_1 \times D_2 \times \ldots \times D_n$, which define feasible combinations of domain values. A solution is an assignment of domain values to all variables which is consistent with all imposed constraints.

The CSP formalization of the MEX-MDP problem is based on a partition of the temporal horizon $\mathcal{H} = [0, H]$ in a set of contiguous windows $W = \{w_1 = [t_0, t_1] \mid t_0 = 0\} \cup \{w_j = (t_{j-1}, t_j] \mid j = 2 \ldots m, t_i \in \mathcal{H}\}$, such that $\cup_{i=1}^{m} w_i = \mathcal{H}$. We consider a set of decision variables $\delta_{i,j}$ that represent the amount of data to be dumped from packet store pk_i in the window w_j. The MEX-MDP contains different kinds of constraints: (a) Given the characteristics of the packet stores the data must be downlink according to a FIFO philosophy; (b) the amount of data for each packet store does not exceed its capacity; (c) a finite amount of data can be dumped in each transmission window (finite transmission rate).

All the proposed algorithms work over two levels of abstraction: (1) the *planning level*, where the whole set of decision variables are instantiated taking into account the different constraints; (2) the *scheduling level*, where a sequence of memory dump operations

is generated over the communication links respecting the constraints imposed over all the windows w_j.
In order to find an optimal solution we choose to realize a heuristic optimization strategy based on *local search* which is able to improve an initial solution given as an input: *Tabu search* [4, 5]. The tabu metaheuristic is founded on the notion of a *move*. A move is a function which transforms one solution into another. For any solution S, a subset of moves applied to S is computed. The result is the *neighborhood of* S. The algorithm proceeds selecting, at each step, the best solution in the neighborhood, with respect to an objective function, till a fixed number of steps are made without finding better solutions.

Figure 1: MEXAR layout

In MEX-MDP the move consists in bringing forward some data (for example data contained in observations with the smallest volume of data) and delaying other ones; this should improve in many cases the objective function (mean turn over time).

Mexar Interactive Functionalities

The MEXAR functionalities that are designed for the users are summarized in Figure 2. As expected the problem solving activity is central in the system. This functionality is guaranteed by the automated services centered on the constraint-satisfaction methodology (CSP) described above.
In the figure 2 we show the flow of control during the use of the functionalities. It is possible to identify an activity that aims generically at defining a single problem. At present it simply consists of loading a problem description from a file, it can be also be replaced by a more complex incremental functionality that could be well coupled with the CSP modeling used. The definition of a problem is followed by its solution according to the different algorithms produced in our work. A different functionality allows to refine the current problem. This activity consists in deleting some of the Payload Operation Requests (PORs) from the associated timelines. This can be useful to experiment different loads on the resources in specific intervals of the solution. This functionality introduces a cycle among these activities that could bring the user to incrementally refine new MEX-MDP problems. As shown we group these functionalities in an interaction layout called Problem Analyzer (see Figure 2).
Once a problem to solve is defined we can attack it with different solution methods. Figure 1 shows an example of a solved instance of MEX-MDP as it is presented to the user.

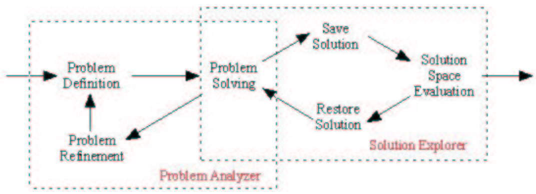

Figure 2: MEXAR Interactive Environments

The availability of a portfolio of problem solving procedures has suggested the idea of involving more deeply the user in the solution process. This has been pursued by creating an environment in which it is possible to save different solutions and, in addition, the user can guide search on how to improve the solutions applying different algorithms. We call this process solution space exploration. This aspect is strictly connected to the availability of evaluation metrics on the solutions as discussed in [2]. The idea behind the solution explorer is the one that the user can generate an initial solution, save it, try to improve it by local search, save the results, try to improve it by local search with different tuning parameters and so on. In this way, it is possible to generate paths in the search space. The user can restore one of the previous solutions and try to improve it with a local search with different parameters, etc. In this way he generates a tree of solutions. This procedure can be repeated for different starting points generating, in this way, a set of trees. Using at the same time the evaluation capability on a single solution and its own experience he can generate different solution series, all of them saved, and, at the end, choose the best candidate for execution.

Bibliography

[1] Cesta, A.; Cortellessa, G.; Oddi, A.; and Policella, N. Interaction Services for Mission Planning in MARS EXPRESS. In *Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space*, 2002.

[2] Cesta, A.; Oddi, A.; Cortellessa, G.; and Policella, N. Using AI Techniques to Solve MEXMDP. Technical Report MEXAR-TR-02-08, ISTC-CNR - Planning and Scheduling Team, Rome, Italy, 2002.

[3] ESA. European Space Agency, MARS EXPRESS Web Site. `http://sci.esa.int/marsexpress/`, 2002.

[4] Glover, F. Tabu Search – Part I. *ORSA Journal of Computing* 1:190–206, 1989.

[5] Glover, F. Tabu Search – Part II. *ORSA Journal of Computing* 2:4–32, 1990.

[6] Montanari, U. Networks of Constraints: Fundamental Properties and Applications to Picture Processing. *Information Sciences* 7:95–132, 1974

[7] Oddi, A.; Cesta, A.; Policella, N.; and Cortellessa, G. Scheduling Downlink Operations in MARS EXPRESS. In *Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space*, 2002.

Author Information

Gabriella Cortellessa[1], **Nicola Policella**[2], **Amedeo Cesta, and Angelo Oddi** ISTC-CNR, Italian National Research Council, Viale Marx 15, I-00137 Rome, Italy, `{corte,policella,cesta,oddi}@ip.rm.cnr.it`

[1] is also Ph.D. student in Cognitive Psychology at University of Rome “La Sapienza”

[2] is also Ph.D. student in Computer Science at University of Rome “La Sapienza” □

ARTICLE

RDPPlan: an Extension of DPPlan for Planning with Interval Resources

Authors: **M. Baioletti, A. Milani, and V. Poggioni**

RDPPLan is a model for planning with quantitative resources. It is based on DPPPlan [1], a planner which uses a non directional search algorithm on the planning graph.

Most models of planning with resources, like [3], [4], [5], [6], [7] and [8], assume that an exact value can model the continuous quantities describing, in the real world, a given resource. In other words these models cannot deal with more realistic situations in which quantities are not known exactly. The RDPPlan model allows one to manage domains where preconditions and effects over quantitative resources can be specified by intervals of values; in addition mixed logical/quantitative and pure numerical goals can be specified.

Instead of initializing a resource with a unique real