A Comparison of Coordinated Planning Methods for Cooperating Rovers

Steve Chien, Anthony Barrett, Tara Estlin, Gregg Rabideau
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, M/S 126-347
Pasadena, CA 91109
{firstname.lastname}@jpl.nasa.gov

ABSTRACT
This paper describes and evaluates three methods for coordinating multiple agents. These agents interact in two ways. First, they are able to work together to achieve a common pool of goals which would require greater time to achieve by any one of the agents operating independently. Second, they share resources that are required by the actions needed to accomplish the goals. The first coordination method described is a centralized scheme in which all of the coordination is done at a central location and the agents have no autonomy at the planning level. The second method performs goal allocation using a centralized heuristic planner and (distributed) planners for the individual agents perform detailed planning. The third method uses a contract net protocol to allocate goals and then (distributed) planners for the individual agents perform detailed planning.

Keywords
Multi-agent coordination and collaboration, multi-agent teams

1. INTRODUCTION
The Mars Pathfinder and Sojourner missions were major successes, not only demonstrating the feasibility of sending rovers to other planets, but also demonstrating the utility of such missions to the scientific community. In order to increase science return and enable new types of observations, new missions are being proposed that employ larger sets of robotic workers. Multiple rovers can behave in a cooperative or even coordinated fashion, accepting goals for the team, performing group tasks and sharing acquired information. Coordinating multiple distributed agents introduces unique challenges for automated planning and other supporting technology. Issues arise concerning interfaces between agents, communication bandwidth, group command and control, and onboard capabilities, all of which will limit the level of autonomy each of the rovers can have.

In our approach, we examine the use of Artificial Intelligence (AI) planning and scheduling in three different control structures to automatically generate appropriate low-level rover command sequences to achieve science goals. In the three approaches, we explore a range of distribution of the planning function ranging from a completely centralized planner to a bidding system in which the planning process occurs on each rover in parallel. Other approaches to multi-agent planning have had various degrees of distribution (Brummit and Stentz 1998; Mataric 1995; Müller 1996).

2. BASELINE SCENARIO
We evaluate the architectures presented in this paper using the following geological scenario. Different Martian rockscapes are created by using distributions over rock types, sizes and locations. Science goals consist of requests to take spectral measurements at certain locations or regions. These goals can be prioritized so if necessary, low priority goals will be deleted first. In each architecture science goals are divided among three identical (simulated) rovers. Each rover has several science instruments onboard, and a solar panel and battery for power. Collected data is immediately transmitted to a lander where it is stored in memory, and only one rover can transmit to the lander at any given time. The lander can also upload data (and simultaneously free up memory) to an orbiter whenever the orbiter is in view.

Formulating plans in this scenario involves dividing goals between rovers in a method that minimizes the amount of driving each rover must perform. Decisions must be made not only to satisfy the requested goals, but also to provide more optimal schedules. When assigning a goal, the architecture must select the best rover for the job and decide the order that each rover will achieve its assigned goals. Decisions are further complicated by the state and resource constraints mentioned above. For instance, communication constraints between the rover and lander may affect when certain science operations can be performed.

All of our architectures require a planner/scheduler to turn abstract science goals into concrete activity schedules. For this problem, we have extended the ASPEN (Fukunaga et al. 1997) application framework, which uses an iterative repair algorithm to search for a conflict-free schedule. To provide more optimal schedules, we have also implemented heuristics based on the Multiple Traveling Salesman Problem (MTSP).

3. MANY ARCHITECTURES FOR COORDINATION
While there are many approaches to coordinating a set of agents, the two most common either treats them as a single master agent directing a set of slaves or treats them as a set of competing peers. In this section we describe these two extreme approaches and an intermediate one.

3.1. Centralized Planning

The master/slave approach to automated planning for multiple agents involves using a single centralized planner. Planning and scheduling for all agents is done with a single ASPEN process on the lander. When planning is complete, the relevant sub-plans are transmitted to each "slave" rover for execution.

This approach has several advantages and disadvantages. One major advantage is that the planning process is conceptually simplified. All commands are sequenced together, allowing any interactions to be easily checked and planned for. A major disadvantage becomes visible when the rovers' environment is somewhat unpredictable. Here the central planner will also have to monitor execution in order to replan activities in response to unexpected failures or fortuitous events. This will involve continuously transmitting large amounts of data to and from the master agent. Finally, this approach has a single point of failure. If the agent running the planner is rendered inoperable, remote planning will not be possible, and command sequences will need to be uploaded from the ground.

3.2. Central Goal Allocation with Distributed Planning

One approach to distributed planning is to include one planner for each agent, in addition to a central planner (Estlin et al. 1999). The central planner develops an abstract plan for all agents, while each agent planner develops a detailed, executable plan for its own activities. The central planner also acts as a router, taking a global set of goals and dividing it up among the agents.

This approach also has its advantages and disadvantages. The obvious advantage is that the planning process is distributed across multiple processors, which reduces the workload on any one agent and allows planning to be done in parallel. Another major advantage is faster reaction time with less communications. With a planner onboard the rovers, there is a tight loop between planning and execution. The major disadvantage of this approach stems from the partitioning of goals and resources from the master to the slaves. Once the goals have been assigned, there is no way for them to be reassigned to different rovers.

3.3. Contract Net Protocol

Migrating the planning/scheduling process onto the rovers leaves a central auctioneer to distribute goals, and the rovers use planning/scheduling to determine appropriate bids for each goal as it arises. This approach is an instance of the *contract net protocol* (Smith 1980), where a manager announces a task to a set of contractors, each contractor bids for it, and the manager awards the task to the contractor with the best bid.

This approach has many of the centralized goal allocation algorithm's advantages and disadvantages. Once again, the planners on the rovers facilitate tight feedback between planning and execution without high communications overhead. The one difference between the decentralized planning approaches involves the information used to partition the goals. Where the previous approach ignored resources on the rovers and partitioned the goals strictly based on expected path distances, the contract net approach partitioned goals based on path distances after taking other rover resources into account.

4. COMPARISONS

The three approaches presented in this paper for coordinating multiple agents have a number of functional differences. One main difference is that both the distributed planning approach and contract net approach can take advantage of parallel processing while the centralized planning approach cannot. Another difference is the degree of autonomy offered by an individual rover with respect to possible replanning. In the centralized planner approach, a failure by a rover that cannot plan would require communication with the central planner before resuming execution. In the case of the distributed planning or contract net approach, if the failure could be planned around locally by the failing rover, such communication would not be necessary.

These approaches were also empirically compared using problems generated from the previously described geological scenario. In these tests we calculated statistics for the number of goals achieved, average distance traveled per goal, and planning computation time (sum and makespan). The contract net approach outperformed the centralized and distributed planning approaches in terms of number of goals achieved. This is because the contract net approach allows the individual planners to each try to achieve every goal – a goal will be deleted only if all rovers cannot plan for it. The average distance traveled per goal was comparable for all of the approaches. Compared to the centralized approach, the distributed planning approach incurred a greater cost in total computation, but the average makespan of the CPU time was lower because it can construct individual plans in parallel. The contract net approach used considerably more CPU time because it invokes each of the individual planners N times for N overall goals. As stated previously, this is likely why the contract net achieves more goals on average.

5. CONCLUSIONS

This paper has described three methods for coordinating multiple agents. These agents interact both by working together to achieve a common pool of goals and by sharing required resources. We compare these approaches and empirically evaluate them using a geological science scenario. For a more detailed version of this paper, please see (Chien, 2000).

ACKNOWLEDGEMENTS

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

REFERENCES

B. Brummitt and A. Stentz 1998. GRAMMPS: A Generalized Mission Planner for Multiple Mobile Robots in Unstructured Environments. In *Proceedings of ICRA-98*.

S. Chien, A. Barrett, T. Estlin, and G. Rabideau 2000. A Comparison of Coordinated Planning Methods for Cooperating Rovers. JPL Tech Report. http://planning.jpl.nasa.gov/papers/rover-comparison00.ps.

T. Estlin, A. Gray, T. Mann, G. Rabideau, R. Castano, S. Chien, and E. Mjolsness 1999. An Integrated System for Multi-Rover Scientific Exploration. In *Proceedings of the AAAI-99*.

A. Fukunaga, G. Rabideau, S. Chien, D. Yan 1997. Towards an Application Framework for Automated Planning and Scheduling. In *Proceedings of the iSAIRAS-97*.

M. Mataric 1995. Issues and Approaches in the Design of Collective Autonomous Agents. *Robotics and Autonomous Systems*, 16 (2-4).

J. Müller 1996, *The Design of Intelligent Agents, A Layered Approach*, Lecture Notes in Artificial Intelligence, Springer-Verlag.

G. Smith. 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, 29(12).