Scheduling Mobile Exploration Tasks for Environment Learning

(Extended Abstract)

Max Korein
The Robotics Institue
Carnegie Mellon University
Pittsburgh, PA 15213, USA
mkorein@andrew.cmu.edu

Brian Coltin
The Robotics Institue
Carnegie Mellon University
Pittsburgh, PA 15213, USA
bcoltin@cs.cmu.edu

Manuela Veloso
Computer Science Dept.
Carnegie Mellon University
Pittsburgh, PA 15213, USA
mmv@cs.cmu.edu

ABSTRACT

Autonomous mobile service robots navigate in their environments in order to perform tasks requested by users. We envision service robots learning about their environment by scheduling exploration tasks in which they seek out new knowledge and using this knowledge to improve the services they offer. We present the Task Graph algorithm, which chooses times for user requests based on the robot's knowledge so as to increase the chance of success, and schedules exploration tasks in between user requests by reducing the problem to a graph search.

Categories and Subject Descriptors

I.2.9 [**Robotics**]

General Terms

Algorithms

Keywords

Learning, Planning, Scheduling, Exploration, Service Robots

1. INTRODUCTION

The CoBots are mobile autonomous service robots that operate in a multistory office building [5, 2]. Users request tasks of the CoBots online, such as delivering a message or finding and fetching an object, to be performed within flexible time windows. Currently, the CoBots have little knowledge of their environment beyond a map. However, the services the CoBots provide could be improved if they stored more environmental knowledge. For example, a message delivery can fail if the recipient's office's door is closed when CoBot delivers its message. If CoBot had a model of when each office door is most likely to be open, it could time message deliveries for when they are most likely to succeed.

In order to learn this information, we envision CoBot scheduling its own "exploration tasks" in addition to user requests. Other work has been done with service robots that acquire knowledge and use it to improve their services, such as Jijo-2, a robot that learns information through conversations with users [1], or Dora the Explorer, which models where to find objects for find and fetch tasks based on past observations [4]. However, these works focus primarily on how the robot can acquire and evaluate knowledge. In this work, we focus on addressing the challenge of managing the robot's time. The robot must perform user requests at the times that maximize the probability of success, and choose how best to use its remaining time for exploration.

Scheduling only user requests within the time windows provided by users is an instance of the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW can be solved using heuristics or approximations, but for CoBot we solve it exactly using mixed integer programs (MIP)[6, 3]. While the MIP approach can handle the number of requests typically made by users (fewer than ten at a time), it is ineffective for scheduling large numbers of user requests and exploration tasks simultaneously.

Thus, when scheduling exploration tasks we cannot rely solely on the MIP approaches CoBot uses to schedule user requests. To this end, we present the Task Graph algorithm, which creates a schedule by first restricting the flexible user-provided time windows of the user requests to improve the expected reward, then schedules exploration tasks in between by reducing the problem to a graph search.

2. THE TASK GRAPH ALGORITHM

The goal of the Task Graph algorithm is to create a schedule of start times for a set of user requests and exploration tasks. Each user request has an interval of constraint in which it must be performed, and all tasks must be completed within a given interval of operation. Additionally, the robot must have sufficient time to travel between tasks, which have specific start and end locations. The robot can evaluate the expected probability of success of any user request or the expected reward of any exploration task (based on the value of the knowledge it expects to gain) at a specific time given the its current knowledge base. The first priority is to maximize the reward achieved by the schedule via user requests. The second priority is the maximize the reward achieved via exploration tasks. Our algorithm has two stages, corresponding to these two goals.

2.1 Maximizing User Request Successes

The first stage of the algorithm seeks to choose start times for all user requests in order to maximize the probability of

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

success. It first selects a time interval for each user request with the maximum average reward, scaled by a power of its length:

$$I_{r_i} = \arg\max_{I \subseteq I_{c,i}} \left[(\text{length}(I)^f) R_{request}(r_i, I|K) \right],$$

where I_{r_i} is the interval chosen for the i^{th} user request, $I_{c,i}$ is the constraint interval for that request, K is the robot's current knowledge base, and f is a flexibility parameter. Initially, f is set to 0, and a set of start times is chosen using CoBot's MIP scheduling algorithm [3]. If no valid set of start times exists, f is increased by $\delta = 0.1$, and the process is repeated until one is found. We assume that a possible set of start times exists for a sufficiently high f.

2.2 Scheduling Exploration Tasks

The second stage of the algorithm schedules exploration tasks in between the user requests to achieve a high exploration reward. The algorithm first finds the intervals of exploration between the user requests during which the robot can schedule exploration tasks. Each interval of exploration also has a start location and end location. For each interval of exploration, the robot then constructs a "task graph," a directed graph with the property that any path in the graph represents a sequence of exploration tasks, and the length of the path is the expected time to perform those tasks. The task graph is constructed automatically. It contains the following nodes:

- "Start" and "end" nodes for each location.
- "Wildcard" start and end nodes with 0 distance to any node.
- "Task" nodes for each exploration task with reward equal to the mean reward of that task over the interval.

The wildcard nodes are used when the interval has no required start or end location. The task graph has these edges:

- From each start node to each task node with length equal to the distance from the start node's location to the task node's start location.
- From each task node to each other task node with length equal to the distance from the first task node's end location to the second task node's start location.
- From each task node to each end node with length equal to the distance from the task node's end location to the end node's location.

The "cost" of a task node is the sum of the length of the edge to that node and the task's duration. Our goal is to find the path from the interval's start location's start node to its end location's end node with length less than the length of the interval. We would like to find the path with the maximum reward, but for larger task graphs this is intractible.

Instead, we use a heuristic search to find a high-reward path. The search begins at the start node and computes the benefit (ratio of reward to cost) of all task nodes. It then selects the task node with the highest benefit, adds it to the path, and continues the search starting from that node. If at any time the end node of the search cannot be reached within the time remaining in the interval, the search backtracks and chooses the next-highest-benefit task node. When no task nodes can be added to the path without exceeding the interval's length, the search is complete, and the tasks in the path are added to the schedule. Each task is assigned the earliest possible start time in order of the path. The schedule is complete when exploration tasks have been scheduled for each interval of exploration.

3. EXPERIMENTAL RESULTS

We tested the algorithm in a simulation of CoBot operating in three floors of the Gates-Hillman Center at Carnegie Mellon University. CoBot was assigned four random message deliveries over 40-minute intervals, and doors had a random probability of being open each minute. Attempting to deliver a message to a closed door resulted in failure. CoBot could explore hallways as exploration tasks, observe doors, and model their probabilities of being open. Results were averaged over 50 trials.

We tested the Task Graph algorithm with both an uninformed uniform exploration reward and an informed reward inversely proportional to the number of observations CoBot had of the doors on a hallway at a given time. After 100 intervals, the average error of the probabilities in the model was 0.312 when using the informed reward, 0.336 when using the uninformed reward, and 0.435 when no exploration tasks were scheduled. Thus, exploring using the Task Graph algorithm improved the robot's model of its environment dramatically, especially when using an informed reward function.

Acknowledgements

This research was partially supported by award NSF IIS-1012733, and by ONR subcontract USC 138803. The views and conclusions contained in this document are those of the authors only.

4. REFERENCES

[1] H. Asoh, N. Vlassis, Y. Motomura, F. Asano, I. Hara, S. Hayamizu, K. Ito, T. Kurita, T. Matsui, R. Bunschoten, and B. Kröse. Jijo-2: An office robot that communicates and learns. *IEEE Intelligent Systems*, 16(5):46–55, 2001.

[2] J. Biswas, B. Coltin, and M. Veloso. Corrective gradient refinement for mobile robot localization. In *Intelligent Robots and Systems (IROS)*, pages 73–78. IEEE, 2011.

[3] B. Coltin, M. Veloso, and R. Ventura. Dynamic user task scheduling for mobile robots. In *Proceedings of the AAAI Workshop on Automated Action Planning for Autonomous Mobile Robots*, August 2011.

[4] M. Hanheide, C. Gretton, and M. Göbelbecker. Dora, a robot exploiting probabilistic knowledge under uncertain sensing for efficient object search. In *Proceedings of Systems Demonstration ICAPS*, 2011.

[5] S. Rosenthal, J. Biswas, and M. Veloso. An effective personal mobile robot agent through symbiotic human-robot interaction. In *Proceedings of AAMAS*, volume 1, pages 915–922. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[6] P. Toth and D. Vigo. *The vehicle routing problem*, volume 9. Society for Industrial Mathematics, 2002.