Using Anytime Planning for Centralized Coordination of Multiple Robots in Real-time Dynamic Environments

Gabriel J. Ferrer, Glenn S. Wasson, James P. Gunderson and **Worthy N. Martin**
Computer Science Department, Thornton Hall
University of Virginia, Charlottesville, VA 22903
{ferrer, wasson, gunders, martin}@virginia.edu

We are investigating the use of planning in multi-robot, real-time, dynamic environments. Each individual robot is considered to be an effector of a centralized planning system. In a real-time environment, it is important to be careful to ensure that the time required to find a good plan does not itself reduce the system's ability to complete its tasks in a timely manner. Our physically situated system coordinates a group of security robots that guard an area against intruders. Our approach uses a variation of the three-level paradigm for robot architectures (e.g. (Bonasso *et al.* 1997) (Brill *et al.* 1998)) in conjunction with an anytime approach to planning (Dean *et al.* 1995) (Zilberstein & Russell 1993). In this domain, the system must ensure that the total amount of time spent planning and then executing the plan is short enough that the security robots can still capture the intruders. Too much time spent planning prevents the security robots from achieving this objective. Our system uses anytime planning to ensure that plans are found in a timely manner. Anytime algorithms return results with a quality proportional to the amount of time the algorithm had available for execution.

Our planner initially assigns patrol routes to the security robots that ensure sensor coverage of the area. As intruders enter the area, our planner assigns some of the robots sequences of intruders to pursue, and plans patrols for the remaining robots to maintain sensor coverage. Once tasks are assigned, the security robots carry out their tasks reactively (Brooks 1986) (Brill *et al.* 1998). Once reactive execution begins, the security robots act independently of the planner until they are assigned new tasks. Each robot reports the current locations of any intruders sensed to the sequencer. The sequencer determines if reassignments are necessary, and if so, reinvokes the planner. Task reassignments are made when new intruders are detected or a security robot is out of range to capture an assigned target.

An anytime approach is used by our planner for robot task replanning. For an anytime algorithm to be effective, it is important to specify a progression of results that the algorithm will return. Our progression is based on the concept of *plan locality*. That is, when a plan fails and needs to be replanned, much of the plan may still be valid and useful as is, and repairing the failure may be sufficient. On the other hand, local failures in a plan can have non-local implications. Our approach initially limits replanning to the part of the plan that failed, while leaving the rest of the plan alone. This constrains the amount of search that the planner must conduct. Once that is done, the amount of the plan to be replanned is enlarged. This allows the planner to search a larger space. This process of iteratively enlarging the part of the plan to be replanned continues until either the planner is interrupted or the scope of replanning has grown to the point that it encompasses the entire plan. The smaller search space of the early phase of the replanning algorithm ensures that there is a new plan available quickly, while the larger spaces searched in the later phases enable plans of higher potential quality to be considered.

Our system currently uses a geometric measure of locality when replanning. That is, it replans for security robots that are within a progressively increasing radius of the plan failure point. Some of the research questions we are investigating include: How can we generalize the concept of locality for replanning in other domains? If the intermediate data structures (e.g. partial plans and world states) of the planning algorithm are preserved, can we use locality as an indexing device to enable us to find partial plans quickly that we can use as starting points for replanning? How should we progress locality when two or more plan failures occur?

References

Bonasso, R. P.; Firby, R. J.; Gat, E.; Kortenkamp, D.; Miller, D. P.; and Slack, M. G. 1997. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental and Theoretical Artificial Intelligence* 9(2).

Brill, F. Z.; Wasson, G. S.; Ferrer, G. J.; and Martin, W. N. 1998. The effective field of view paradigm: Adding representation to a reactive system. *Engineering Applications of Artificial Intelligence* 11:189–201.

Brooks, R. A. 1986. A robost layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* RA-2(1):14–23.

Dean, T.; Kaelbling, L. P.; Kirman, J.; and Nicholson, A. 1995. Planning under time constraints in stochastic domains. *Artificial Intelligence* 76(1-2):35–74.

Zilberstein, S., and Russell, S. J. 1993. Anytime sensing, planning and action: A practical model for robot control. In *Proceedings of IJCAI-1993*, 1402–1407.