CS 396 Special Topics in Artificial Intelligence
Project Deliverable 4

Ari Wilson

March 13th, 2007

Progress

Many difficulties were encountered in the last week, implementing more advanced agents within the framework of my resource scenario. As agent functionality and modules began to become more elaborate, I have been attempting to manage this complexity by merging the existing implementation portion of my project into an Eclipse project. This effort, thanks to Sean's help, is now complete. Eclipse is an advanced Java IDE that is widely considered to be to Java what Visual Studio is to C++.

Spring Break, however, has not put a hamper on the theoretical side of the project. I have begun to develop a scheme for modeling my problem in terms of reinforcement learning when thinking of my scenario as a *Markov decision problem*, a single-agent stochastic game. Other types of learning will be described more fully as they are sketched out within the context of my scenario, including belief-based reinforcement and no-regret learning with associated modifications.

Schedule

Difficulties with the implementation part of this project mean that I am behind schedule. My revised schedule for the rest of the semester is:

- *Before project demonstration*: Agents implementing the learning methods listed in this deliverable will be implemented and tested within the simulation world thus far developed.
- *Project demonstration*: Refined versions of the agents developed thus far will be demonstrated, as well as the statistic-capturing features of my system.
- *Project deliverable 5*: This deliverable will consist of final renditions of a number of agents as refined from deliverable 3 and those developed after 4 with updated documentation and initial performance statistics.
- *After deliverable 5*: This work will consist of performance evaluations and analysis, a presentation detailing exactly how the project went, and a final write-up of lessons learned and achievements over the course of the project.

Distributed planning

Distributed planning is only peripherally related to my project. Any of the main types, including centralized planning with distributed execution, distributed planning with centralized execution, or distributed planning with distributed execution, fundamentally deals with problems with more

complex notions of success than are present in my simple scenario, designed to show the effects of different methods of learning with adjusted parameters.

My reasoning here is related to the notion of utility in my repeated game. The only measure of utility in my project is the final distribution of resources over the complete set of agents, measured either as a group or individually. In the individual utility case, no agent has incentive to respond positively to any plan that does not increase its own utility over its own plan. But since an agent would have chosen the plan that it believes will maximize its overall utility in the first place, distributed planning is useless!

Things are a bit more complicated when discussing group utility. Since my agents do not have any guarantees with respect to other agents' behaviors, there is no way to guarantee that any group cohesion or competence exists or can be acted upon, as is necessary for the successful completion of any plan. Put another way, some agents may be less willing or completely unable, as in the case of my simple agents, to cooperate with others. Predicting group utility in this simple scenario can suffer from the horizon effect, where the results of plans are greatly affected by the actions of their most unpredictable components, the less-cooperative agents.