CIS099, Intelligent Game Agents – Project report

Ryan Gormley, Eliot Kaplan, Fabian Peternek
Teamname: Team BWAAAAAAMP

December 9, 2011

Prelude

The following report shall describe our efforts to create an artificial intelligence, that conquers the world as known to the ants or at least wins the competition. We will start by an explanation of our first plan and then detail the method we used in the end.

Act I: The plan

The basic idea was the following: We realised, that reinforcement learning works well if all the parameters are set correctly. Setting those parameters however is quite difficult, without some deeper insight into the problem. This is especially true for the reward function, where one reward that is out of line can unintentionally bias the learning process into one undesired direction. We therefore wanted to try the following: We would use reinforcement learning (or rather q-learning) to train the bot, but we would first learn the reward function by using another search strategy. To do this we first intended to use a genetic algorithm approach, but this would have taken too long to yield a good result. A good reward function was therefore to be found by grid search.

Act II: The approach

The first step in getting this plan to work was to create features, that incorporate the anthills, which were previously not used. We therefore created features for the distance of an ant to the closest known own and enemy ant hills, and how much food is stored in the hill. The latter one however is only an approximation, as the game unfortunately does not communicate this number to the bots. Furthermore the q-learning algorithm was adapted to get a parameterized input for the reward function and the other important variables and a trainer for this parameters was created using a grid search approach.

Act III: The problems

At this point we could start to train our bots, but unfortunately had to realize, that the training does not work. What went wrong? Likely more than one thing, but the most important one is: The game has changed considerably. The fact, that the only way to get points is now to destroy an enemy ant hill makes it difficult to evaluate the results of training, where we used the difference in score to discern, which bot was better. However as long as no bot accidentally moved onto an enemy hill, there was no way to tell, if one bot did better than another. At this point "won the most games" might have been a better approach of evaluation.

Another problem is, that our reward function had no information about the hills. We could define a negative reward for loosing the own ant hill, but we did not think of incorporating the destroying of enemy hills into our reward function.

Act: IV: The solution

Due to the changed scoring we could not get the idea of searching for a good reward function to work in time. We therefore decided to tailor the rewards by hand again and run the q-learning on that. However we did think of rewarding the ants for destroying enemy hills this time. This approach finally lead to a working bot.

Act V: The Tournament

As our bot was finished just in time for the final tournament, we could not train a lot. Furthermore, the tournament would be our evaluation of the bot. Considering those prerequisites it actually did not do bad. It lost one game and almost won another one, only loosing because it timed out. With some more tweaking we are quite confident, that this bot would be working pretty well.

Epilogue

The last two pages detailed our ants way to world domination, even though it did not quite manage to get there. We described our original plan for the bot, which parts worked and which parts did not work. We supplied some some assumptions why these parts did not work and then detailed how we solved the problems in time. The report is concluded by a short evaluation of its performance, showing, that the bot was not terrible, even though some more tweaking would be pretty helpful.

Analysis

Rating

We ran a 100 game tournament against greedy bot, and our bot lost most of the matches. our bot won 17 games and lost 83. However, based on the total score, our bot managed to destroy the enemy hill in all 17 victories, and greedybot happened to destroy our hill in 8 games.

Strengths

Our original plan showed a lot of promise. By grid searching for a reward function rather than making one by hand, we could ensure that the behaviors most conducive to victory would be rewarded. With more time, we'd have implemented a better scoring metric to make the grid searched rewards work properly.

Our actual bot excelled in a few ways - it correctly learned to keep ants fairly widely spread out in order to best explore the map. It also did a pretty good job of avoiding unprofitable encounters with enemies.

Weaknesses

Our bot's biggest weakness was just lack of training. We lost a ton of time to the gridsearching, so when we finally got qlearning off the ground, it was the night before the competition. And then our overnight training went bad while we slept, so we were forced to enter a severely undertrained bot into the tournament. We are convinced that with a full 50 training games under its belt, our bot could have been a formidable opponent.