Bartley Gillan
CS 344M
05/01/2008

Final Report: Building a Winning Team named Agent Kings

1. INTRODUCTION

The purpose of this report is to show my final project results of developing a winning team of agents for the Robocup soccer simulator tournament to be held at the end of the semester. My team of agents are named the Agent Kings. The starting point for my team has been the UVA 2003 code base and I do not have any additional team members. From this code base, I have implemented some of the ideas used in the ATT-CMUnited-2000 Robocup team and also made use of the existing UVA code. These ATT-CMUnited-2000 ideas are found in the article, "An Architecture for Action Selection in Robotic Soccer" by Stone and McAllester [1].

My general strategy in developing a competitive, winning team of agents was to leverage the UVA code base as much as possible while implementing what ideas I could from the ATT-CMUnited-2000 team. I think implementing some of the features and skills used by the ATT-CMUnited-2000 team will give me an edge in winning the class tournament, because these skills lead to good global team behavior. This global team behavior would be noticeably more improved than the majority of the students' teams in Programming Assignment four. The teams that focus on a highly specialized aspect of the Robocup soccer problem will be at a disadvantage compared to teams that focus on good global team performance. While these more specialized teams may do one task much better than an average Robocup student team, the team designers will inevitably give less attention to the other tasks necessary to make a good team. I predict this skill imbalance will favor those teams that focus on improving all tasks to a satisfactory level. In the end, the team that scores the most goals will win the tournament. One cannot accomplish this by having a good shooter, but subpar teammates. All eleven agents on the team must work well, and do so together, to exhibit good team performance and score many goals.

2. PROJECT FOCUS AND STRATEGY

It would have been nice to implement all of the ATT-CMUnited-2000 features for my team of agents, but limited time and resources made this task impractical. Instead, I implemented the features that give the most noticeable team performance improvement with the least coding difficulty. I chose to focus on developing the offensive skills and strategy of my team, because the base UVA defense functionality seemed to work well enough. This strategy is aggressive in trying to maximize the number of goals as possible, but as the old adage goes, "the best defense is a good offense." However, if facing a highly skilled offensive opponent, the UVA defense may yield too many goals to the opponent team.

The first sign of the aggressive offense is the 3-3-4 formation or "SwissBolt" that my team takes at all times [2]. This formation has one more attacker than midfielder and defender. "On attack, the bolt [has] a 3-3-4 shape complete with an attacking center half, with all the players, including the three-man fullback line, moving well upfield" [2]. When the opposition has possession, the entire formation pulls back to defend. This formation has its advantages in creating shot opportunities as long as a good job is done keeping the ball on the offensive side of the field.

3. BALL POSITIONING

When my team is in possession of the ball, the movement of the ball towards strategic offensive locations is done by evaluation of a ball position value function. These strategic locations are, not surprisingly, at the opponent's goal and the adjacent corners. The three locations will be referred to as ball attraction points, and they are involved whenever my team has possession of the ball and is moving down the field. This ball position value idea comes from Stone and McAllister's work [1]. A value function calculates the value of the ball's position at any point on the field. As the ball moves closer to the opponent's end-line, the ball position value increases linearly. The distance is also calculated from the ball to each of the three attraction points. A value inversely proportional to the relative distance is assigned for each of the three ball attraction points. A weighted sum of the values for the opponent's end-line, corners, and goal is calculated with all weightings being equal. The equal weightings was the first thing that was tried. This seemed to work well enough, so no additional tuning was done. The final value sum is then used to lead the ball possessing agent to dribble or pass to a teammate in an area where the ball's position will have a higher value. This value is used in the evaluation of different passing options. Given the location of every teammate on the field, a position value can be computed for each agent. Then, the agent possessing the ball can simply pass the ball to an agent that has a higher computed position value than itself, which will be an agent that is closer to the opponent's goal or adjacent corners. It is not clear whether or not that pass will be successful based on this information alone, because the teammate receiving the pass may be well guarded.

4. CROSS SHOT STRATEGY

Another idea borrowed from the ATT-CMUnited-2000 team was the ability to perform cross shots. This means when the agents are in control of the ball, they will try to move the ball strategically into one of the offensive corners if the front of the goal is heavily guarded. This strategy will be used to set up cross shot opportunities from the corners to teammates in an area in front of the goal. The authors of ATT-CMUnited-2000 had at least two teammates move to the front of the goal whenever a teammate had possession of the ball in an offensive corner. As described in the author's paper [1], "...if several opponents are nearby, the fact that the ball ends up bouncing around in front of the goal can often lead to unpredictable, effective shots."

The 3-3-4 formation used has two forward wings and two center forwards. If one of the forward wings possesses the ball, it will generally move toward the opponent's nearest corner. While the wing with the ball is moving towards the corner, the rest of the formation is moving upfield and the two center forwards are moving up towards the penalty box and goal. The wing then kicks the ball towards the goal area once it is within 20 m of the corner or 30 m of the goal. The hope is that either the wing will make the shot past the goalie or the ball will be deflected off an opponent so the two center forwards can get a shot on the goal. The values of 20 m from the corner and 30 m from the goal were arrived at by trial and error starting with smaller distances and then increasing them until the positioning and timing of the cross shot improved. This cross shot strategy is also used when my team gets a corner kick. In practice against the UVA Trilearn 2003 team, the goalie blocks the majority of the wing cross shots, but on occasion the shot will sneak past the goalie or the center forward will get a good deflection shot to score.

5. BALL CONTROL ALGORITHM

The algorithm I use for controlling each individual agent, excluding the goalie, is not too complicated, but requires attention. All agents will shoot the ball towards the nearest goal corner if it is within 30 m of the goal. This value was obtained via trial and error. If my team has a corner kick, the agent will clear the ball towards the front of the goal. If the opponent did a back pass and my team has a kick from the corner of the penalty box, the agent will shoot the ball towards the nearest corner of the goal. If none of the aforementioned cases are met, the agent will try to pass to a teammate according the ball position value mentioned in Section 3. It will do this by first calculating its own ball position value. Then it will calculate the ball position value for the closest two teammates relative to the agent. If the closest teammate has a higher ball position value than the current ball possessing agent, the agent will fast pass to it. Otherwise, it will check if the second closest teammate meets this constraint and fast pass to it if it does. If neither of the closest two teammates has a higher ball position value than the agent's current position, or if the agent did not perceive the teammates, then it will try to outplay the nearest opponent. This outplay skill was given in the UVA code base.

First, the agent determines the ball position value of eight equidistant samples on a circle surrounding the agent of radius 7 m. Then the agent tries to move the ball and beat its nearest opponent to the location of the sample with the maximum ball position value. Eight samples was the first thing tried and the 7 m radius appeared to perform the best of trying 3 m, 5 m, and 7 m. If the agent does not think it can outplay the nearest opponent, it will clear the ball towards the nearest half of the offensive side of the field. In this last case, if the agent did not perceive a nearest opponent to outplay, it will kick the ball towards a random corner of the goal.

If a forward wing has possession of the ball, the algorithm is somewhat different. The forward wing will dribble fast towards the opponent's nearest corner, unless it is nearer to the goal. Once the forward wing is within 20 m of the opponent's corner or 30 m of the goal, it will take a cross shot towards the goal corner. These distances were reached via trial and error. When an agent does not possess the ball, the UVA Trilearn base algorithm is used. That is, the teammate that is fastest to the ball will go to intercept it. Otherwise, it will go to a strategic position based on the ball, own goal, nearby opponents, and the team formation. The goalie of the team has the same default behavior as the UVA Trilearn 2003 team. I felt it worked sufficiently well, so I focused on other parts of the team.

6. EXPERIMENTAL RESULTS

I conducted an experiment of having my Agent Kings team play fifty games against the UVA Trilearn 2003 team. The games were split up so that my team would play twenty-five games on one side, and twenty-five on another. This was done not only out of fairness, but also because I initially suspected a bug in my code where my team did not perform as well when on the left side. The results appear in the table below.

Table 1. Agent Kings vs UvA Trilearn 2003 results of 50 games

	Wins	Losses	Draws
Agent Kings (left)	9	12	4
Agent Kings (right)	10	9	6
Agent Kings (total)	19	21	10

As one can see, there is a one game difference from the win/loss/draw record being dead even. This is not a substantial enough difference to say that one team is any better than the other. When my team played on the left side, they had three more losses than wins, whereas when my team played on the right side, they had one more win than losses. This difference is also not substantial enough to confirm my suspicion that there is a left-side-related bug in my code as I originally suspected.

A two-sample, two-tailed, equal variance student's t-test was conducted on the point scores of the fifty games to see if my team exhibits a statistically significant positive score difference. The null hypothesis is that my changes made no significant improvements to the team performance and the average score difference would be 0. The average positive score difference computed of the fifty games is 0.06. Looking at the t-distribution table for 98 degrees of freedom at 95% confidence, my t-value would need to be greater than 1.98 for the results to be statistically significant. The t-value for my score data is 0.83, so the null hypothesis is confirmed and my changes made no significant improvements to the team performance.

7. CONCLUSION

In conclusion, the focus of my project was to develop a winning team of agents for the Robocup soccer simulator tournament. I created an offensive-minded team leveraging the UVA code base and implemented select features found in Stone and McAllester's ATT-CMUnited-2000 Robocup team [1]. Even though the experimental results showed that my changes made no significant improvements to the team performance, my Agent Kings team can match the UVA Trilearn 2003 team in a large number of games played. Whether my team will perform similarly against the classmates' teams of agents remains to be seen, but will no doubt be interesting and entertaining.

REFERENCES

[1] Peter Stone and David McAllester. "An Architecture for Action Selection in Robotic Soccer." In *Proceedings of the* Fifth *International Conference on Autonomous Agents*, pp. 316–323, ACM Press, New York, NY, 2001.

[2] J. Bluem, "Evolution of Systems of Play, Part I", Accessed Apr. 2008, http://www.nscaa.com/subpages/20060331154402100.php.