ATT-CMUnited-2000: Third Place Finisher in the Robocup-2000 Simulator League

Patrick Riley[1], Peter Stone[2], David McAllester[2], and Manuela Veloso[1]

[1]Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
pfr,mmv@cs.cmu.edu
http://www.cs.cmu.edu/ pfr, mmv

[2]AT&T Labs — Research
180 Park Ave.
Florham Park, NJ 07932
dmac,pstone@research.att.com
http://www.research.att.com/ dmac, pstone

1 Introduction

The ATT-CMUnited-2000 simulator team finished in third place at Robocup-2000. It was one of only three teams to finish ahead of the previous year's champion, CMUnited-99. In controlled experiments, ATT-CMUnited-2000 beats CMUnited-99 by an average score of 2.6–0.2.

ATT-CMUnited-2000 is based upon CMUnited-99 [4], which in turn is based on CMUnited-98 [5]. In particular, it uses many of the techniques embodied in the previous teams, including: Flexible, adaptive formations (Locker-room agreement); Single-channel, low-bandwidth communication; Predictive, locally optimal skills (PLOS); and Strategic positioning using attraction and repulsion (SPAR).

The ATT-CMUnited-2000 team explores several new research directions, which are outlined in the sections below.

2 On-Line Planning for Set Plays

CMUnited-99 uses a variety of "set-plays" for the various dead-ball situations (goal kick, free kick, corner kick, etc.). However, the set plays have very little variation among executions. ATT-CMUnited-2000 uses dynamic, adaptive planning for set-plays. The coach for ATT-CMUnited-2000 makes a new plan for player and ball movement at every offensive set play.

A plan is represented as a Simple Temporal Network, as introduced in [2]. A Simple Temporal Network (or STN for short) is a directed graph that represents the temporal constraints between events. For example, some of the events used here are "start pass", "start goto point", and "end goto point." The temporal constraints are used to represent the parallelism inherent in having multiple agents as well as allowing the agents to detect failures during plan execution.

Muscettola *et al* have described a "dispatching execution" algorithm for STN execution[3]. The algorithm allows easy and efficient propagation of temporal constraints through the network. Our execution algorithm extends the dispatching execution algorithm to the multi-agent case. We use the dispatching algorithm (unchanged) to maintain information about time bounds for executions of events. Every cycle, the agents monitor conditions on plan execution and work towards bringing about the execution of events in the appropriate order.

In order to create plans, we use models of opponent movements. Because of the short time span in a simulated soccer game, we decided to fix models ahead of time and to choose between them during the game. Selecting a model should be much faster than trying to actually create a model online. Conceptually, we want an opponent model to represent how an opponent plays defense during setplays. We expect that a wide range of decision-making systems of the opponent can be roughly captured by a small set of models. In order to handle the uncertainty in the environment and to allow models to represent more information than just a simple predicted location, we use probabilistic models. Given the recent history of the ball's movement (from the start of the set play for example) and the player's initial location, the models give, for each player, a probability distribution over locations on the field.

In order to use a naive Bayes update to select between models during the game, we make the following assumptions. While the assumptions are probably not completely correct, we expect that we can still get reasonable model selection with these assumptions.

1. The players movements are independent. That is, the model may generate a probability distribution for player x based on everyone's starting locations. However, what the actual observation is for player x (assumed to be sampled from this probability distribution) is independent from the actual observations of the other players.
2. The probability of a particular starting position and ball movement are independent of the opponent model. This assumption is questionable since the ball movement (i.e. plan) generated depends on the opponent model.

However, the model which collectively makes the best prediction may not be the model we would like to use for planning. We want to use the model that predicts the players closest to the ball the most accurately because the planning will depend mostly on players closest to the ball. The model that most accurately captures those players should give us the best plans from the planner. Further, we would like to have a smooth transition in how the players the players affect the probability of a model as their distance changes. We have created a weighting scheme which can be applied to the naive Bayes update in order to achieve these properties.

The planning process largely consists of solving a path planning problem with moving, probabilistic obstacles (where the obstacles are the opponents). We use a hillclimbing approach which balances the safety of a plan with the benefit obtained (in terms of final ball location) if the plan succeeds.

For defensive set plays, the coach assumes that the plan which the opponents execute will be similar to previous set plays. The coach instructs players to go to locations which would be good if the ball and player movements were identical to previous plays. The coach does this by drawing a boundary around the current ball position and tracking where the ball crosses this boundary.

3 Computing Interception Times

A significant difference between ATT-CMUnited-2000 and its predecessors is that ATT-CMUnited-2000 introduces the concept of a *leading pass*. In the past, most teams have only considered passing the ball directly to a teammate. However, there is a potentially large advantage to be gained by passing the ball in such a way that the teammate can "run on" to the ball.

In ATT-CMUnited-2000 the player holding the ball considers hundreds of possible kick directions and kick powers and, for each hypothetical kick, calculates the time for each other player (both teammate and opponent) to reach the ball. This entire calculation is done once each simulation cycle whenever the player is holding the ball. Forward simulation methods that reason about discrete simulator cycles, such as those used by past champion teams Humboldt [1] and CMUnited [5], are not computationally efficient enough to evaluate hundreds of potential passes (although they are very effective and tractable for evaluating small numbers of passes). ATT-CMUnited-2000 abstracts the simulation to continuous time and uses a modified Newton's method to efficiently approximate earliest interception times.

For an arbitrary ball position B_0, initial ball velocity V_0 and receiver position R_0, we want the least time required for the receiver to reach the ball assuming that time is continuous, the receiver can run at a fixed velocity of V_R starting immediately, and the ball continues in its current direction with instantaneous velocity after time t given by $V = V_0 e^{-t/\tau}$. The receiver velocity V_r is taken to be the maximum player velocity in the simulator — one meter per second. The velocity decay parameter τ is adjusted to match the simulator at the discrete simulation times — τ is about 2 seconds. The position of the ball after time t, denoted $P(t)$, can be written as follows.

$$P(t) = B_0 + V_0 \tau (1 - e^{-t/\tau}) \tag{1}$$

The distance between the initial receiver position and the ball position at the interception time must equal the distance the receiver can travel. This condition can be written as follows where $||X||$ denotes the length of vector X.

$$V_r t = ||R_0 - P(t)|| \tag{2}$$

To formulate the problem in a way appropriate for Newton's method we first define $g(t)$ as follows.

$$g(t) = ||P(t) - R_0|| - V_r t \tag{3}$$

Condition (2) can now be written as $g(t) = 0$. Unfortunately, this condition often has three different roots. To find only the smallest root we set t^0 to be the d/V_r where d is the distance from R_0 to the line defined by the point B_0 and the vector

V_0. This gives that t^0 is no larger than the smallest root. The standard Newton's method update sets t^{i+1} to be s^{i+1} as defined by the following equation.

$$s^{i+1} = t^i - g(t^i)/g'(t^i) \tag{4}$$

We now modify this update rule to ensure that if t^i is no larger than the smallest root then t^{i+1} is also no larger than the smallest root. Let $U(t)$ be the unit vector in the direction from R_0 to $P(t)$. We can rewrite equation (4) as follows.

$$s^{i+1} = t^i + g(t^i)/(V_r - e^{-t^i/\tau} V_0 \cdot U(t^i)) \tag{5}$$

Our modified update rule is the following where s^{i+1} is defined by equation (5).

$$t^{i+1} = \begin{cases} s^{i+1} & \text{if } V_0 \cdot U(t) < 0 \\ t^i + g(t^i)/(V_r - e^{-t^i/\tau} V_0 \cdot U(s^{i+1})) & \text{if } V_0 \cdot U(t) > 0 \text{ and } g'(t^i) < 0 \\ t^i + g(t^i)/V_r & \text{otherwise} \end{cases}$$

A proof is available upon request that the sequence t^0, t^1, t^2, ... converges to the smallest root of (2).

4 Conclusion

In empirical tests, ATT-CMUnited-2000 beat CMUnited-99 by an average score of 2.6–0.2 over the course of 33 10-minute games. There are two major improvements to ATT-CMUnited-2000 which helped it achieve these results and place third at RoboCup2000. The online coach is used to make a new plan for each setplay. The planning algorithm uses a model of the opponent's movements in order to evaluate generated plans. Given a set of models before the game, the coach selects which model best describes the current opponent. The players also use a fast numerical technique in order to evaluate hundreds of kicking options when they have the ball. Thus, they are able to execute successful leading passes, rather than only passing directly to teammates.

References

1. H.-D. Burkhard, M. Hannebauer, and J. Wendler. AT Humboldt — development, practice and theory. In H. Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, pages 357–372. Springer Verlag, Berlin, 1998.
2. R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.
3. N. Muscettola, P. Morris, and I. Tsamardinos. Reformulating temporal plans for efficient execution. In *KR-98*, 1998.
4. P. Stone, P. Riley, and M. Veloso. The CMUnited-99 champion simulator team. In Veloso, Pagello, and Kitano, editors, *RoboCup-99: Robot Soccer World Cup III*, pages 35–48. Springer, Berlin, 2000.
5. P. Stone, M. Veloso, and P. Riley. The CMUnited-98 champion simulator team. In Asada and Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, pages 61–76. Springer Verlag, Berlin, 1999.