Coevolving Quidditch Players Using Genetic Programming

Student: Michael Ahern

Advisor: Sergio Alvarez

6th May 2005

Table of Contents

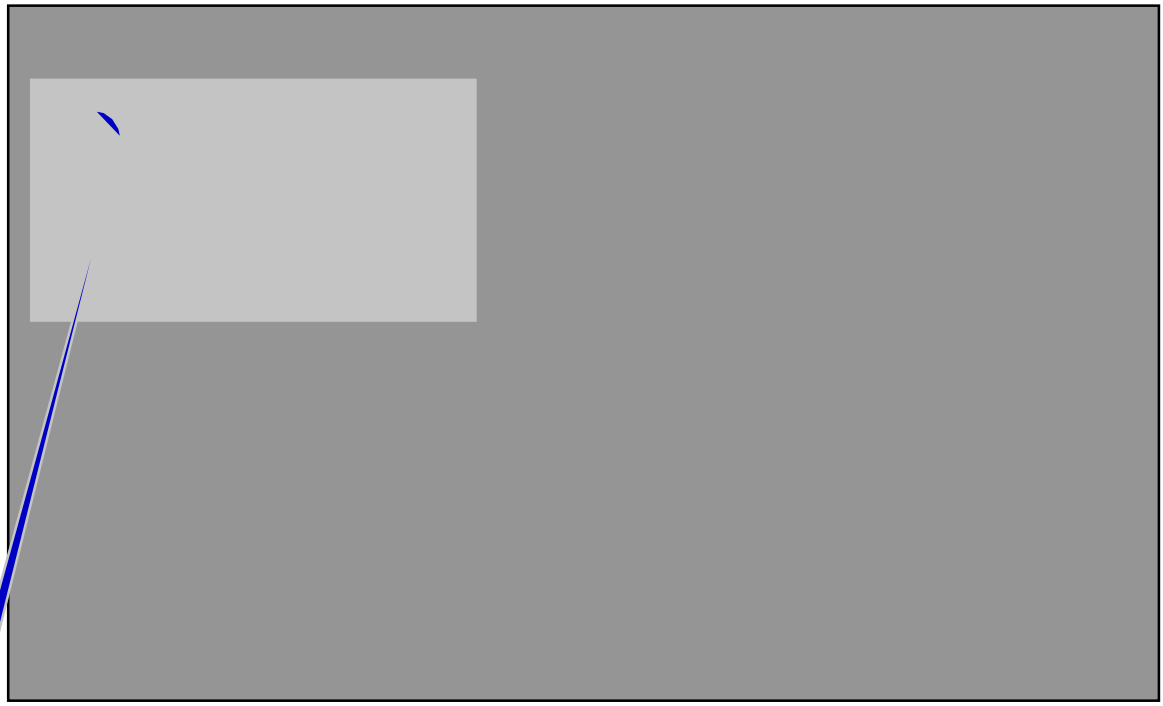
List of Tables

Abstract

Chapter 1: Introduction

For those who have neither read the books nor seen the films, Quidditch is the most popular sport played in J. K. Rowling's series of books. The game resembles a fusion between basketball and soccer played on flying brooms. Two teams,

yellow they receive a higher fitness score. After selection, the clone() function preferentially copies two red, two blue, and one yellow individual into the next generation. Then the crossover() operator splits the red and blue objects in half, mixing one half of each to produce two purple objects. Finally, the mutate() operator takes a blue object and somehow alters its "genes" to produce the light blue color seen in generation n+1.



1.2 Genetic Programming

Genetic programs (GP's) are exactly like GA's except in that their representation is an actual computer program rather than a binary string or an Lab color value. Typically, following the work of Koza (Koza, 1992), LISP-like function trees are used to represent these programs (see figure 4). Rather than being an arbitrary program written in C, the function set, following from the problem, is chosen by the user. The challenge then with this model is to select a set of functions and non-terminals that will be flexible enough to represent the problem solution, while being constrained enough to make the search tractable. Take, for example, the simple case trying to evolve the function: "sin(x) +

sqrt(x^2 + y)". Given that problem one would need a function set that included "sin", "sqrt", "x", "y", "cos", and so forth, in order to represent the solution. With more complicated problems, the function set is not quite as obvious. Hence, the task of the genetic programmer is to somehow provide a flexible solution that includes enough hints to make the problem tractable.

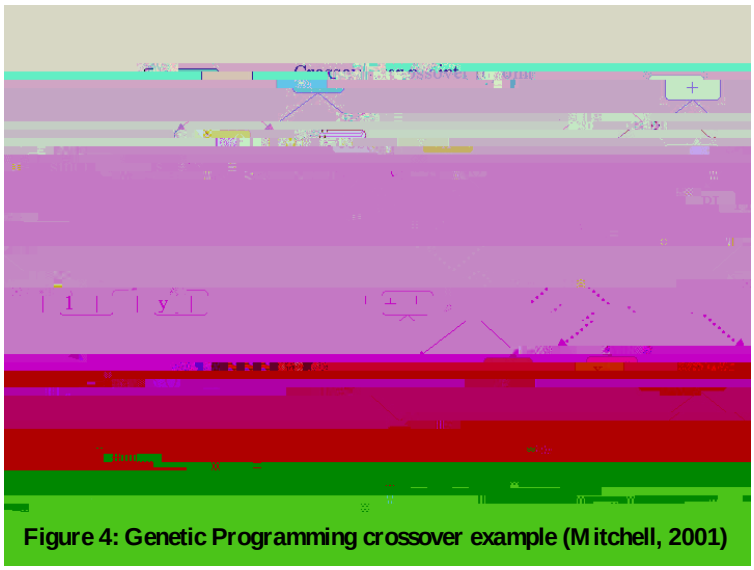


Figure 4: Genetic Programming crossover example (Mitchell, 2001)

1.3 Previous Research

1.3.1 RoboCup Software

The RoboCup tournament was started in the mid-nineties with the goal of developing a human-competitive team of robotic soccer players by 2050. Each year, the Robo World Cup has been held, pitting teams of either physical robots or simulated soccer players against each other.

The sT -0.E (c) -47 (c) -47(he) -47 (r) -38 ( ) -41 (s) -62(ai) -13 (m)7-33 (u) -71 (l) -13 (a) -47 (t)5T -0.E (r) -38 ( ) -41(ai

"dribble" to more complex actions such as "blocking the goal". This implementation was fairly successful pointing towards later work in the same domain. The two-program tree methodology presented here also served as the basis for the implementation presented in this paper.

1.3.2 Virtual Witches and Warlocks

Addressing Spector, Moore, and Robinson's proposal, Raphael Crawford-Marks (Crawford-Marks, 2004) attempted to develop a Quidditch GA based on earlier work headed by Lee Spector to develop a Quidditch simulator. The simulator developed implemented the full game of Quidditch as described in
(Whisp and Rowling, 2001), including possession time limits, fowls, and score tracking. The evolver created teams using a "stack-based" language called . In addition to player teams, a set of "ball" was coevolved as well. Each player team consisted of seven

While Crawford-Marks had some success with this implementation, the system

Chapter 2:

from the top of their respective trees, action is taken. In the case of move-related vectors, the vectors are vectorized to (targetPoint – currentPoint) and fed to the “thrust” function

The Crossover Operator

DTree crossover is restrictive in the sense that it can only occur at type-same crossover positions. What this means is that only Boolean returning sub-trees can be switched with Boolean returning sub-trees, only integer returning sub-trees with integer returning sub-trees, and so forth.

Crossover-Point Selection Algorithm

```
SelectCross = on (mom, dad | mom, dad are DTree's)
      momP = rand(1:|mom|)
```

2.2 DTree Functions and Rationale

DTree contains a combination of conditional (if-V, if

2.4 The Quidditch Simulator

Chapter 3: Run Results

3.1 Run Setup

Due to time constraints and bugs, only one evolutionary run of the system was completed. The system was tested on a Dell Inspiron 500m⊇ laptop with a 1.3Ghz Pentium M processor and 640 MB of system memory. With this setup, the run involved populations of 100 individuals and took approximately 16 hours. See table 3 for a detail of run parameters.

Table 3: Run Parameter Descriptions

Parameter	Meaning	Value
Ngen	Number of generations to run.	100
Popsize	Size of TreesGenome population	100
baseTime*	Number of seconds to allow games to run for.	15.0

Figure 14: Generation 0 – Carry Behavior

Figure 15: Generation 0 – Go to Ball Behavior

By the intermediate generations (15-20), all the players learned to go for the ball. At this stage, players generally threw the ball immediately at nothing in particular (figure

Gradually, between generation 20 and 100, players learned to hold the ball longer before throwing it (figure 17). Occasional spikes resulted when the player was even

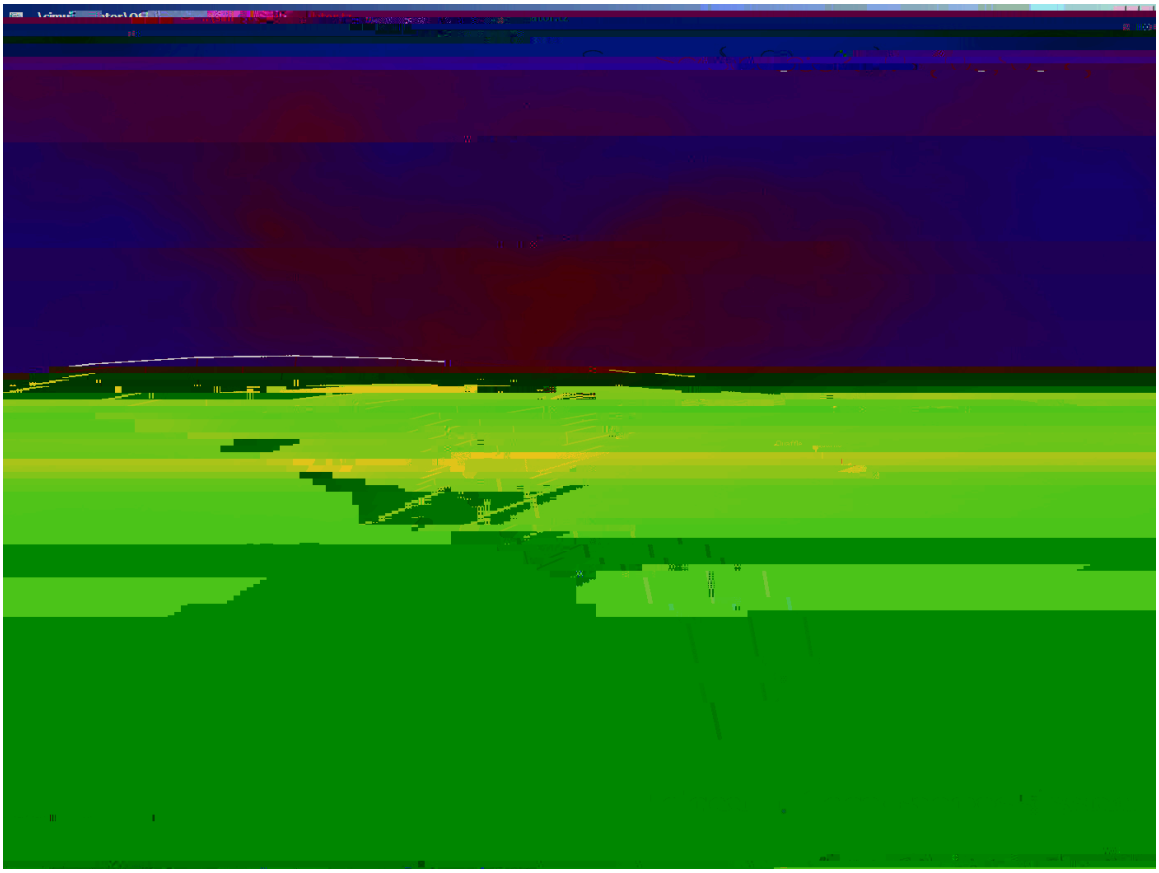


Figure 18: Generation 100 – Shooting on Goal

4.1 Future Work

Future work can be roughly divided into two categories, either involving the simulator or involving the Evolver. Simulator improvements center around increasing realism while evolver improvements concern bug fixes and AI performance (Crawford-Marks, 2004).

4.1.1 Evolver Improvements

As mentioned in the beginning of this chapter, the most pressing issue to be addressed by future work is the timing bug. Beyond that, a number of the behavioral functions, most notably the generate-throw function seems to perform less than optimally. Although there was little discussion of these functions, in terms of game states, there is a class of probabilistic game-state functions, such as “throw-near-goal-if”, that were based on (C * (1 / dist)) approximations of throw accuracy. In addition to improving the throw function, more careful analysis of these probabilities should be carried out.

Beyond these bugs, the distinction between the and trees is

4.1.2 Simulator

Architecture – As it stands now, the simulator does not control the player input as tightly as could be desired. In order to increase the control of the input, the simulator should probably be moved to a client/server model. Although there is a bit of network overhead in this model, by having an always-running simulator, approximately 1.5 seconds out of a 4.5 second simulation run is spent on system initialization, this delay could be removed. Thus, given that m

Appendix A: Quidditch and the Hampshire College Simulator

NOTE: The following section is from Raphael Crawford-Marks' division III thesis, pp. 14-24. The text was modified slightly due to table numbering changes, but is essentially as is for the purposes of reference.

The Game of Quidditch

Balls

The Quaffle is a round, inflated leather ball, painted red. In the early 18th century, witch Daisy Pennifold enchanted the Quaffle to fall as though sinking through water. The "Pennifold Quaffle" is still used today. In 1875, another enchantment was added to the Quaffle; a "gripping charm" allowing Chaser to easily keep hold of the Quaffle with one hand.

The Bludgers started out as enchanted rocks, sometimes carved into the shape of a ball. Modern Bludgers are heavy iron balls, 10 inches in diameter.
Bludgers are bewitched to chase the player closest to them. Therefore Beaters must try to knock Bludgers as far away from their teammates as possible.

The Golden Snitch is a walnut-sized golden ball with thin, translucent wings. It is fast, highly maneuverable and semi-intelligent, employing all its abilities to avoid being caught by either Seeker.

Players

The Keepers are like soccer goalies. They hover near their own goals to fend off shots from opposing Chasers. Keepers have all the same abilities as Chasers, so they do not exist as a distinct player class in the Quidditch Simulator. Rather, if Keeper-like behavior turns out to be adaptive, then one or more Chasers can evolve to act as a Keeper.

The Chasers are somewhat analogous to soccer forwards. They can hold on to the Quaffle, pass it back and forth, and throw it at the opposing team's goal hoops. Normally there are only three Chasers per team, but in the Quidditch Simulator there are four because there is no Keeper.

The Beaters defend their teammates from the Bludgers. They are equipped with wooden bats to knock Bludgers away from their teammates.

The Seeker is tasked with capturing the Golden Snitch. They are usually the fastest and most agile player on the team. When the Golden Snitch is caught, the capturing team is awarded one-hundred and fifty points, and the game ends.

Gameplay

Quidditch is played over an oval-shaped field five hundred feet long and one hundred and eighty feed wide. This is called the Pitch. At each end of the pitch are three goal hoops. does not specify the height of the goal hoops. In the Quidditch Simulator, they are 15 meters high.

At the start of the game, all players are grounded on their team's half of the pitch. The referee whistles, and the balls are released at midfield. The Quaffle is thrown into the air by the referee. At this point the Quidditch match has begun, and does not end until the Snitch is caught or both team captains consent to end the game.

mo (t26 (.) 32 (g) 49(.) 7 (r) 60 (a) 26 (n)49(g) 7 (t) 7 (.) 49(.) 32 (r) 49(.) 7 (me)t26 (.) 32 (a) 49(i) 60 (a)26 (a

As soon as the referee whistles the start of the game, the Keepers rush to their respective scoring areas to defend the goals (there are no Keepers in the Quidditch Simulator, if this behavior is adaptive then hopefully it will be adopted by one of the four Chasers). The Chasers lift off and scramble after the Quaffle. The Beaters track the Bludgers and assume strategic positions to defend thei

shapes. Collision detection is elegantly handled by Breve, whereas detection of the Quaffle passing through a hoop would have been much more difficult to program.

Each team has one extra Chaser. This is because Keepers (as described in) are simply Chasers that elect to defend instead of attack. This being the case, there was no reason to create a separate Keeper class. If it is adaptive to have a Keeper then hopefully one or more Chasers will evolve defensive behaviors. Indeed, some very simple defensive behavior was observed during each of the three large evolutionary runs in which one of the four Chasers would fall back to the center goal and orbit it, occasionally intercepting shots from the opposing team.

The value of catching the Snitch has also been modified. Instead of being worth one-hundred and fifty points, capturing the Snitch is worth 10+(2*score) (up to 150) points for the capturing team. This change was made because teams were evolving to only chase the Snitch, completely ignoring the Quaffle.

Name	Applies to	Description
Blagging	All Players	Seizing the opponent's broom tail to slow or hinder
Blatching	All players	Flying with intent to collide
Blurting	All players	Locking broom handles with a view to steering opponent off course
Bumphing	Beaters only	Hitting Bludger towards the crowd, necessitating a halt of the game as officials rush to protect bystanders. Sometimes used by unscrupulous players to prevent an opposing Chaser scoring
Cobbing	All players	Excessive use of elbows towards opponents
Flacking	Keeper only	Sticking any portion of anatomy through goal

The Simulator Architecture

QuidditchTeam does for players. The ScoreTracker class is also a subclass of Abstract. The ScoreTracker performs many of the same functions that a scoreboard would at a basketball or football game. It keeps track of the score, which team has possession, the state of the game (The list of game states can be seen in Table 3.4), and how much time is left. ScoreTracker also writes the fitness score of each team to a file after the game is over.

meters, mean 0, standard deviation (distance/5) gaussian noise is added to each component of the sensed vector location.

Appendix B: Full DTree Quidditch Function Listing

Logical and Integer Functions

Key: t – throw, i – integer, b –

Boolean State Functions

(opp-closer)	B	if opponent closer to the ball, else true.
(mate-closer)	B	if a teammate is closer to the ball, else false.

(opp

Vector Input Functions

Function Syntax	Returns	Description
(home)	V	A vector to my home.
(home-mate i)	V	A vector to the home of teammate THIRD( )
(ball)	V	A vector to the ball.
(goal i)	V	A vector/throw to the goal THIRD( ).
(closest-goal)	V	A vector/throw to the goal.

Vector Action Functions

Function Syntax	Returns	Description
(block-goal)	V	A vector to the closest point on the line segment between the ball and the goal I defend.
(away-mates)	V	A vector away from known teammates, computed as the inverse of Sigma{m{vect teammates} frac{max - ‖ m ‖}{‖m‖} * m }
(away-opps)	V	A vector away from known opponents, computed as the inverse of Sigma{m{vect teammates} frac{max - ‖ m ‖}{‖m‖} * m }

Throw / P-Throw Functions

Function Syntax	Returns	Description
(far-mate i t)	T	A throw vector to the most offensive-positioned teammate who can receive the ball with at least a frac{ + }{12} probability. Otherwise return

(near-mate i t)

Bibliography

Crawford-Marks, R. _____. Thesis. Hampshire College, 2004. 10 Apr. 2005 <http://alum.hampshire.edu/~rpc01/div3.pdf>.

Goldberg, D. , ,
(1989). Addison-