AI Magazine Volume 19 Number 3 (1998) (© AAAI)

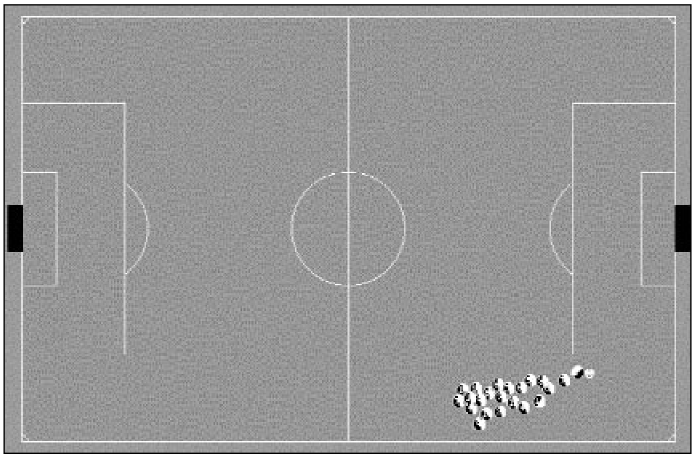
Coevolving Soccer Softbots

The University of Maryland's RoboCup simulator team consisted entirely of computer-evolved players developed with *genetic programming* (Koza 1992), a branch of evolutionary computation that uses natural selection to optimize over the space of computer algorithms. Unlike other entrants that fashioned good softbot teams from a battery of relatively well-understood robotics techniques, our goal was to see if it was even possible to use evolutionary computation to develop high-level soccer behaviors that were competitive with the human-crafted strategies of other teams. Although evolutionary computation has been successful in many fields, evolving a computer algorithm has proven challenging, especially in a domain such as robot soccer.

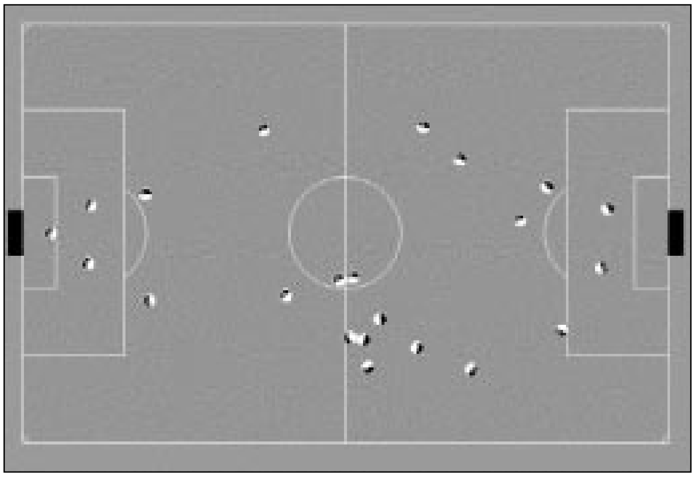
Our approach was to evolve a population of teams of Lisp *s*-expression algorithms, evaluating each team by attaching its algorithms to robot players and trying them out in the simulator. Early experiments tested individual players, but ultimately, the final runs pitted whole teams against each other using coevolution. After evaluation, a team's fitness assessment was based on its success relative to its opponent. This fitness score determined which teams would be selected to interbreed and form the next generation of algorithms.

The RoboCup soccer simulator makes evolutionary computation extremely difficult. The simulator gives noisy data, limited sensor information, and complex dynamics. Most problematic is that the simulator runs in real time; even at full throttle, games can take many seconds or even minutes. Unfortunately, evolving a team of 11 soccer players can require hundreds of thousands of evaluations; so, in the worst case, a single soccer evolution run could take a year or more to complete.

To keep the number of evaluations to a minimum, we severely limited the population size, which demanded special customizations to prevent the population from converging to a suboptimal strategy. We also cut down the number of evolved algorithms by grouping players into squads, with one algorithm to a squad, or using one single algorithm for the entire team (Luke and Spector 1996). We performed runs for both strategies; by the time of the competition, the single-team strategies had better fitness. To further speed up runs, evaluations were run in parallel on an ALPHA supercomputer cluster.



"Everyone Go after the Ball": A Popular but Troublingly Suboptimal Early Team Strategy.



Soccer Teams Eventually Learn to Disperse Themselves throughout the Field.

Because we had only one shot to evolve teams, we cannot make rigorous scientific claims about population development. Nonetheless, an admittedly anecdotal observation is still interesting. After a hesitant start, most early teams soon began to learn the worrisome suboptimal kiddy soccer strategy: Everyone go after the ball, and kick it to the goal (top figure). Thankfully, eventually players learned to hang back and protect the goal and, ultimately, disperse through the field to provide better coverage (bottom figure).

By the end of the final runs, the computer had produced teams that, as appropriate, passed to teammates, blocked the ball, protected different parts of the field, and tried to stay open. Our submission was surprisingly successful, beating its first two hand-coded competitors before succumbing. Hopefully this and other experiments will show that evolutionary computation is ready for a number of problems that previously have only been the purview of human ability.

– *Sean Luke*