# Creating Intelligent Agents through Shaping of Coevolution

Adam Dziuk
Department of Computer Science
The University of Texas at Austin
Email: adziuk@cs.utexas.edu

Risto Miikkulainen
Department of Computer Science
The University of Texas at Austin
Email: risto@cs.utexas.edu

*Abstract*—Creating agents that behave in complex and believable ways in video games and virtual environments is a difficult task. One solution, shaping, has worked well in evolution of neural networks for agent control in relatively straightforward environments such as the NERO video game, but is very labor-intensive. Another solution, coevolution, promises to establish shaping automatically, but it is difficult to control. Although these two approaches have been used separately in the past, they are compatible in principle. This paper shows how shaping can be applied to coevolution to guide it towards more effective behaviors, thus enhancing the power of coevolution in competitive environments. Several automated shaping methods, based on manipulating the fitness function and the game rules, are introduced and tested in a "capture-the-flag"-like environment, where the controller networks for two populations of agents are evolved using the rtNEAT neuroevolution method. Each of these shaping methods as well as their combinations are superior to a control, i.e. direct evolution without shaping. They are effective in different and sometimes incompatible ways, suggesting that different methods may work best in different environments. Using shaping, it should thus be possible to employ coevolution to create intelligent agents for a variety of games.

## I. Introduction

In many games there are non-player characters (NPCs) intended to be humanoid in both form and function. In order to make the game interesting, it is important that such characters behave in interesting and efficient ways. However, creating such behavior is difficult: Most commonly it is scripted, which can lead to predictable and inflexible agents. An alternative approach is to have agents adapt variability: such adaptability can make the game experience more diverse and therefore more interesting for the players [14]. While it is conceivable that such agents could be constructed simply by letting them play against many human players, currently such experiences are not extensive enough and learning is not fast enough to make this approach possible. However, there is another possibility: having the learning agents play themselves, resulting in much more situations from which to learn. This is the main idea for using coevolution to construct intelligent game agents.

In coevolution, a population of agents evolve together with another population such that their fitnesses are in some way tied together. When these agents are directly competing with each other, for example in symmetric head-to-head contests, or in an asymmetric predator-prey relationship the goal is to establish a coevolutionary "arms-race" [2]: one population's improvements forces the other population to improve, which in turn forces the other population to improve even more, and so on. Coevolutionary arms-races can in principle lead to discovery of complex, original behaviors that can make the game dramatically more interesting and challenging.

However, such an arms-race is not easy to establish, and even when it does happen, it may lead to behaviors that are undesirable or uninteresting. Although much work has been done to encourage evolution towards better behaviors, most of it is focused on maintaining the ability to beat the earlier opponents [1, 9, 10, 15]. This paper explores another possibility: shaping the coevolutionary process directly by modifying the fitness function and the environment during evolution.

In a "capture-the-flag"-like environment, where two teams compete to gather the most coins, several automated such methods are evaluated. All shaping methods result in better performance than not shaping. Interestingly, some of their combinations are not as powerful as the methods individually, suggesting that they are utilizing incompatible dimensions of the task, and further that different methods may work best in different environments. The main conclusion is thus that shaping coevolution is a powerful and versatile technique to create intelligent agents for complex games.

## II. Related Work

The individual techniques of coevolution and of shaping have been known for quite sometime and there has been significant research done on both subjects. For instance, Rosin and co-authors [10, 11] introduced competitive fitness sharing, shared sampling, and Hall of Fame methods of enhancing competitive coevolution, and Stanley et al. developed the NEAT technique that preserves earlier behaviors in the network topology evolution [15]. Coevolution is also theoretically well understood in terms of game-theoretic solution concepts [3, 5]. Similarly, shaping is well understood as an abstraction of shaping in biology [12]. It is widely used in machine learning, from supervised learning [4] and reinforcement learning [8] to evolutionary computation [6, 13], where it is often shown to make it possible to solve problems where direct learning fails.

However to our knowledge, shaping has not been applied to coevolution. Shaping (as conceived in the above approaches) requires a human experimenter to design a schedule of changes
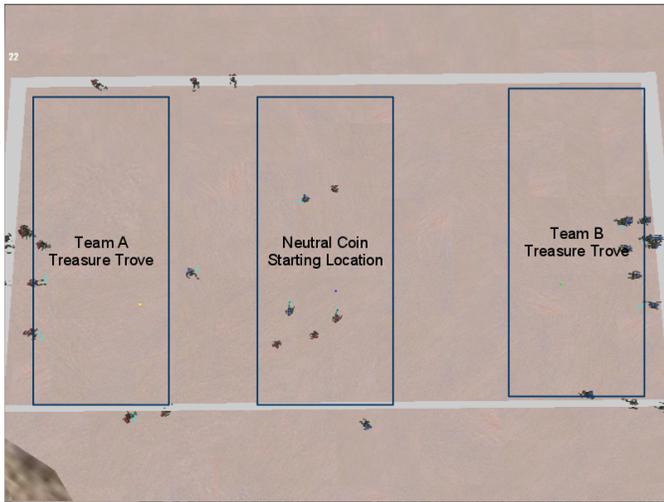
Fig. 1. **The Capture-the-Coins Environment.** In this bird's-eye view of the game video display, the world is inclosed in a (lightly colored) fence, the coins appear as small golden dots, and the two teams as blue and red humanoid robots. The added annotations indicate the spawn/collection areas for each team as well as the neutral area in the center where the coins are initially located. The objective is to collect more coins than the opponent team. This game lends itself to shaping through fitness (i.e. rewarding offense vs. defense) and the environment (i.e. how many agents are needed around the coin to capture it).



Fig. 2. **A sample sensor sector.** In this image, the sensors in one of the eight sectors in the front $180^o$ are shown. Two of the sectors are $90^o$ wide and sense friendly coins; the remaining six are $30^o$ wide and sense enemy coins. In addition, each sector senses the number of friends and number of enemies in the sector. The agent also has one sensor that detects whether the agents is withing the attack/defend range of a coin.

to the task, whereas coevolution itself can be seen as an automatic shaping method: the fitness depends on the behavior of the opposing population, which is gradually becoming more proficient in a coevolutionary arms-race. It seems that manually shaping the fitness or the environment at the same time may interfere with this automatic process, rendering it ineffective, or unnecessary.

In contrast, the experiments in this paper show that shaping can indeed improve the power of coevolution. On the other hand, they also show that the different methods of shaping may indeed be incompatible. It is therefore important to understand how they work m in order to select a method most appropriate for the current task. The environment where such an analysis was undertaken will be described next.

### III. THE CAPTURE-THE-COINS ENVIRONMENT

The goal was to design an environment where offensive and defensive agents could be easily and clearly defined, and thus the shaping process would be transparent for both humans and coevolution. One way to achieve an arms-race in such an environment, then, is to have the teams alternate between attacking and defending behaviors.

These considerations led to the Capture-the-Coins environment, shown in fig 1. It consists of two teams, spawning on opposite ends of a field, attempting to collect as many coins as possible. Three coins are initially placed at random locations in the center area of the field; when a coin is captured, it is moved to the team's own end, but still remains available for the other team to capture. In order to capture a coin with, there must be more attackers than defenders within its vicinity. If an agent is located in the enemy trove (as defined in figure 1),
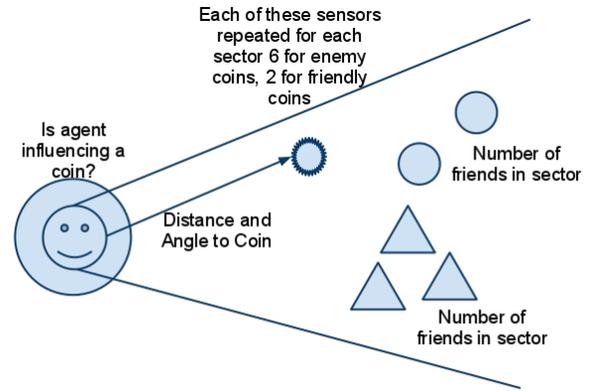
it is deemed to be behaving offensively; if it is located in its own trove, it is defending.

The agents have 34 sensor through which it sees the coins, teammates, and enemies. The sensors are arranged into eight sectors facing the front $180^o$ of the agent; a sample sector is shown in figure 2. Two of these sectors are $90^o$ wide and sense the angle and distance to the closest friendly coin. Overlapping with these sectors are six sectors that sense the angle and distance to the closest enemy coin (because attacking coins requires more accuracy than defending them). In addition, in each sector there is one sensor for number of enemies detected in that sector and one for the number of friends detected in that sector. The agent also has a sensor detecting whether or not the agent is currently within the attack/defend range of a coin (which was 5/6th of the distance that the agent can move in a single time step), and a bias input. These sensors were determined in preliminary experiments to be as small a set as possible (allowing for fast evolution), while still allowing good performance in the task.

As output actions, the agents can turn up to 90 degrees left or right, and they can move either forward or backward at full speed, or not at all. The outputs of the networks determine the actions of the individual agent.

The fitness calculation is a linear combination of offensive behavior rewards (i.e. those involved in capturing coins in the enemy trove), and defensive behavior rewards (i.e. those involved in defending the teams coins in its own trove from capture by the other team). The reward for capturing is given once the coin is captured, and split among all agents within the attack/defend distance. The reward for defending is given every time step that the coin is being attacked by one or more enemy agents, but prevented from being captured; this reward is again split among all the defenders. In this

manner, individual fitnesses were obtained for each member of the population, measuring how well they contributed to the success of the team, allowing evolution to generate agents that cooperated well.

The environment was implemented in the OpenNERO research platform (http://opennero.googlecode.com; [7]). The agents were controlled by neural networks, evolved with the rtNEAT (real-time NEAT; nn.cs.utexas.edu/?rtneat; [14]). This method was chosen because it allows the entire population of neural network to be evaluated in the game at once, replacing each of the agents with their offspring one at a time while the game is going on. Evaluation is therefore highly efficient and the game is continuous.

## IV. APPROACH

In the capture-the-coins environment it is very common for non-shaped teams to converge to uninteresting and non-diverse solutions, such as always defending. Further, these solutions are relatively stable: While the specifics are improved with time, new strategies or even variances of the existing strategies do not typically arise. Shaping can be used to counteract this problem, leading the agents towards a more effective solution.

There are two key factors that can be shaped. The first factor is the fitness function. This aspect can be very useful for encouraging the development of offensive vs. defensive behavior, i.e. by changing the weights on these two components in the fitness function. The second factor is the difficulty of capturing coins, i.e. the percentage of attackers among all agents in the vicinity of the coin that is needed for capture. If this value is raised higher than the initial 50%, more than one attacking agent will be needed to overcome each individual defender; conversely, a single agent can often successfully defend a coin. Similarly, if the value is lowered from 50%, a coin can often be captured by an individual attacker; conversely, defending it requires teamwork. By changing this ratio, it is thus possible to encourage either individual performance or effective teamwork, both in offense and in defense.

In this paper, these two aspects are utilized systematically through automated shaping methods. Three such methods were conceived and tested both individually and in combination with one another (where possible). Unless otherwise specified, the both offensive behavior and defensive behavior contribute 50% to the total fitness, and a coin is captured if more than 50% of the nearby agents are attackers.

In the first method, Alternating Fitness, every three generations the fitness function of one team switches from 75% reward for offensive behavior and 25% reward for defensive behavior to a 25% offensive and 75% defensive reward, while the other team switches in the opposite direction. In this way, at any given point the two teams have opposing fitness functions, encouraging an arms-race where one team has several generations to improve their defense while the other team is being heavily encouraged to attack them.

In the second method, Dynamic Fitness, the fitness weights are adjusted based on the number of coins in the team's possession, as shown in table I for the situation where there

| Captured Coins | Defensive Reward Weight | Offensive Reward Weight |
|---|---|---|
| 3 | 100% | 50% |
| 2 | 75% | 50% |
| 1 | 50% | 75% |
| 0 | 50% | 100% |

TABLE I

**Rules for Dynamic Fitness.** WHEN THE TEAM HAS FEW COINS, ATTACKING IS REWARDED MORE, WHEREAS WHEN IT HAS MANY COINS, DEFENDING IS REWARDED MORE, THUS REWARDING BEHAVIOR THAT WOULD MOST HELP THEIR SCORE.

are three coins on the field. The behavior that would improve the team's situation more is rewarded more: If most coins do not belong to the team, capturing is more heavily rewarded, while when the team already has the most coins, defending is more heavily rewarded.

In the third method, Alternating Rules, every three generations the rules for capturing a coin are alternated such that for one team more than 66% of the agents surrounding an enemy coin must be attackers in order for the coin to be captured, but for the other team, only 34% suffices. This difference is enough to change the dynamics of the most common challenge, i.e. when there is only one defending agent: either one or three enemy agents are necessary for the coin to be captured. Therefore this shaping strategy alternately encourages individual behavior and team behavior, resulting in more successful teams overall.

Each of these three shaping strategies were evaluated experimentally alone and in combination to determine how effective they each were at improving evolved agent behavior.

## V. EXPERIMENTAL SETUP

Six different shaping methods were tested:

1) Alternating Fitness: As described above.
2) Dynamic Fitness: As described above.
3) Alternating Rules: As described above.
4) Alternating Fitness + Alternating Rules: Applying both methods at the same time.
5) Dynamic Fitness + Alternating Rules: Applying both methods at the same time.
6) Control: No shaping at all; the fitness weights and capture requirements were set to 50% for both populations and left the same for the entire simulation. Three different mutation rates were used to test the effect of increased diversity: baseline (same as in the other methods), 3×, and 6× higher rate.

Each algorithm was tested by running it for 100 generations, which turned out in preliminary experiments to provide enough time for shaping to interact with evolution. During each generation, each of the 40 members of the population was evaluated in a game 150 time steps long. A copy of the current population was saved in every ten generations. After each such run, a round robin tournament (called individual run tournament) was run where every saved population was tested against every other twice (once for each end of the field), and
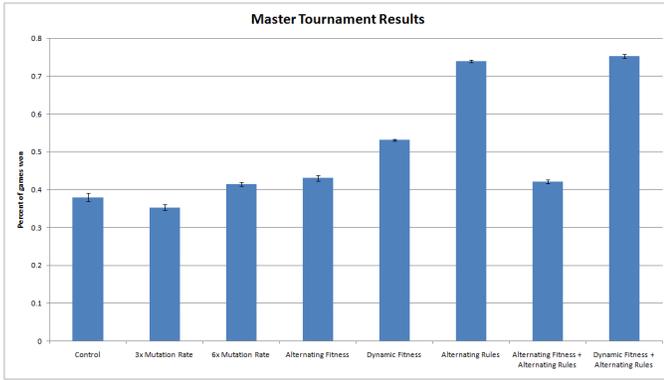
Fig. 3. **Master Tournament Results**. The percentage of games won by each method (i.e. the winners of 10 individual run tournaments of each method) averaged over five master tournaments. The actual numbers are shown in figure II. Alternating Rules, and especially when combined with Dynamic Fitness, results in significantly better performance than the other methods, although any form of shaping is better than none at all. The results thus demonstrate that shaping can be used to make coevolution more powerful.



Fig. 4. **Individual Run Tournament Results**. Percentage of wins of each saved population in the 100-generation run is shown for each method. The curves generally trend up, suggesting that progress is made over evolution. However, there's significant difference in the win percentage of the first saved population (listed in detail in table III), suggesting that it can be used as an early check of whether the particular coevolutionary arms-race works well in the particular task.

once against itself. Each game in the tournament was run for 1200 time steps (eight times longer than during evolution), in order to get a more accurate evaluation. Victory in each game was determined by the average number of coins held by the team over all time steps.

The team that won the most of these games (or a team from the earliest generation in case of a tie) went on to a master tournament, where the winners of 10 runs of each algorithm were tested against each other. This tournament was run five times and the results averaged. Although the results of these five tournaments were very similar, there were slight differences due to the stochastic nature of the environment (i.e. initial coin placement).

## VI. RESULTS

While the results of the master tournament demonstrate that the shaping methods are effective, the individual run tournaments lead to insights into how they do it.

### A. Master Tournament Results

Figure 3 and table II show the results from the master tournament runs. From these results it is clear that any form of shaping at all is a significant improvement over the control. Further, Alternating Rules shaping is most effective, followed by Dynamic Fitness shaping, and combining the two results in a slight improvement on average. Alternating Fitness on its own is better than when combined with Alternating Rules, however. The reasons for these results will be discussed below, however the most important conclusion is that they strongly support the idea of shaping as a way to make coevolution more powerful.

### B. Individual Run Tournament Results

The master tournament shows that while shaping is better than not, different techniques have different power, and are not always compatible, which also suggests that they may be differentially effective in different tasks. It would therefore be
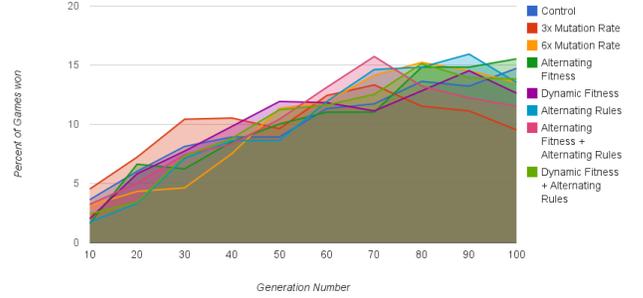
desirable to have a simple way to evaluate which techniques are working well in a given task, instead of having to run a large master tournament.

Indeed, figure 4 suggests that such a test is possible. In this figure, the win percentage is shown as a function of time for the different shaping methods. Regardless of the method the curves trend upward, i.e. generally the later the population was saved, the higher the number of wins it will have in the tournament. However, there is considerable difference in how poorly the first saved population does in this tournament, and it correlates well with how well the method performs eventually (table III). Indeed, strong performance of an early population means that coevolution is unable to maintain an arms-race. Therefore, an individual run tournament can be used to estimate how well each method is likely to do in shaping coevolution. One exception is the Alternating Fitness method which performs much more poorly than expected; the reasons for this anomaly will be discussed in the next section.

## VII. DISCUSSION

The main result is that shaping works well in establishing an arms-race in coevolution. Moreover, automated shaping along the fitness and environment is a more practical approach to the problem then one might first expect. As a matter of fact, in informal experiments it proved to be much more effective than shaping by a human designer.

Several interesting questions arise from the results as well: Why exactly is shaping so effective? Why is Alternating Rules the most effective (and why does combining it with dynamic fitness only improve it slightly)? Why did the Alternating Fitness method produce such effective behavior against its earliest saved population but still perform poorly overall? Why is adding Alternating Fitness to Alternating Rules detrimental? From the answers to these questions, general guidelines for shaping can be inferred.

| Generation | Control | 3× Mutation | 6× Mutation | Alternating Fitness | Dynamic Fitness | Alternating Rules | Alternating Fitness + Alternating Rules | Dynamic Fitness + Alternating Rules |
|---|---|---|---|---|---|---|---|---|
| 0 | 37.30% | 34.78% | 42.39% | 44.34% | 53.08% | 73.52% | 42.45% | 74.65% |
| 1 | 39.37% | 35.09% | 41.32% | 42.64% | 53.40% | 74.15% | 41.57% | 74.91% |
| 2 | 36.92% | 36.48% | 41.51% | 42.58% | 53.46% | 74.03% | 41.76% | 75.79% |
| 3 | 38.81% | 34.53% | 41.07% | 43.21% | 52.96% | 73.96% | 42.08% | 75.91% |
| 4 | 37.67% | 35.53% | 41.38% | 42.70% | 52.83% | 74.21% | 42.70% | 75.47% |
| AVERAGE | 38.01% | 35.28% | 41.53% | 43.09% | 53.14% | 73.97% | 42.11% | 75.35% |

TABLE II
MASTER TOURNAMENT DETAILS. THE WIN PERCENTAGES ARE LISTED FOR EACH METHOD IN EACH OF THE FIVE MASTER TOURNAMENTS, RUN WITH DIFFERENT RANDOM NUMBER SEEDS. THE AVERAGES ARE SHOWN GRAPHICALLY IN FIGURE 3.

| Generation | Control | 3× Mutation | 6× Mutation | Alternating Fitness | Dynamic Fitness | Alternating Rules | Alternating Fitness + Alternating Rules | Dynamic Fitness + Alternating Rules |
|---|---|---|---|---|---|---|---|---|
| 10 | 3.6% | 4.5% | 3.1% | 1.6% | 2% | 1.7% | 3.2% | 2.4% |

TABLE III
PERFORMANCE OF THE FIRST SAVED POPULATION IN THE INDIVIDUAL RUN TOURNAMENT. THE PERCENTAGE OF WINS OF THE FIRST POPULATION, SAVED AT GENERATION 10, IS SHOWN FOR THE DIFFERENT METHODS (THE SAME DATA IS GRAPHED ON THE LEFT SIDE OF FIGURE 4). THE FEWER WINS THERE ARE, THE MORE PROGRESS IS MADE IN LATER GENERATIONS, SUGGESTING THAT THE METHOD WORKS WELL IN THIS DOMAIN.

## A. Effectiveness of shaping

Why is shaping better than not shaping? Analysis of individual games gives some insight into this question. First, the more successful teams had a sparse, relatively even coverage of the enemy trove as well as their own at all times. Indeed, due to the game mechanics, ideally the team should be evenly spread out in the opponents trove, with a percentage of the team staying behind to defend the captured coins. This strategy works well for both the agents individually and for the team as a whole.

In contrast, the control team was relatively dense in these areas. This behavior suggests that these teams consisted of rather similar agents, i.e. that diversity had been lost during evolution, which in turn makes progress in evolution difficult. Diversity can be brought back by almost any change in the fitness function or the environment, which is exactly what the shaping methods do. More different strategies are employed, which in turn makes more efficient exploration possible, and results in discovery of better strategies in the long run.

Interestingly, increasing diversity blindly, through the higher mutation rates, does not result in a similar effect. There's more variation in the population, but apparently very little of it is of any use. Instead, shaping diversifies through behaviors that have been found effective under slightly different circumstances, and therefore form a better foundation for evolution than random diversity.

## B. Effectiveness of Alternating Rules

Given the above observation, it is possible to conjecture why Alternating Rules were so effective: Changing the environment results in strongly increased diversity. Indeed while changes in fitness may encourage behavioral shifts in one way or another, environmental changes do something far more drastic and immediate – they fundamentally change what types of strategies will be successful. Evolution must branch out farther and discover fundamentally different strategies, such as those where one agent captures a guarded coin and those where three agents collaborate to do it.

Changes in the fitness function to reward the diversity may still slightly enhance the evolution, since such pressures are orthogonal to environmental pressures. However, alone they are not as effective: offensive and defensive behavior are still composed of similar components (of teaming up and approaching coins), whereas individual vs. team attacks require a fundamentally different approach (relying on different sensors and different timing). Environmental shaping is therefore more effective in promoting diversity, and thereby more powerful evolution in general, at least capture-the-coins and similar tasks.

## C. Illusion of progress with Alternating Fitness

Why are the late populations of Alternating Fitness so much more powerful than the early populations, yet the algorithm as a whole does not perform as well as others? A possible explanation, based on observing individual games, is that they have learned relatively good defensive behaviors, but have not learned to attack. Therefore, they will easily lose to later teams which have even rudimentary success in both.

Indeed, good defense is easier to establish than offense. It requires simply staying around close to the coins in the teams trove, whereas offense requires more mobility, i.e. traveling to the other side of the field, selecting a lightly defended coin, and traveling from coin to coin as they are captured. Even though fitness alternates between favoring offense and defense, defense will be learned earlier. Early generations therefore tend to know only how to defend, whereas later generations also have some offensive ability. Even though such ability is not strong enough to do well in the master tournament, it is enough to beat the early generations. Other shaping methods reward both behaviors more equally, making the performance of early generations a good indicator of their overall power.

### D. Incompatibility of Alternating Fitness and Alternating Rules

Even though the progress in Alternating Fitness is not as strong as it first seems, it is better than not shaping at all. Then why is it detrimental to add Alternating Fitness to Alternating Rules? A possible answer is that because both rely on time for pacing, there are some detrimental interactions. In particular, there are times when a particular task is both very well rewarded and very easy to do (such as individual defense), and evolution quickly converges the population to a simple solution from which it is difficult to continue.

Dynamic Fitness avoids this problem by emphasizing behaviors that actually matter. It would not spend time on individual defense unless there are many coins to defend, in which case that behavior actually makes sense—but only until the situation changes again. It is therefore less likely to lose diversity, and instead in combination with Alternating Rules discovers well-performing solutions slightly more efficiently than either method alone.

### E. General shaping guidelines

The above observations suggest several general guidelines for shaping coevolution:

1) Automated shaping of coevolution is highly effective, often more so than human-shaped coevolution.
2) Shaping is effective through encouraging diversity.
3) Changing the environment is more effective in encouraging diversity than shaping fitness (at least in domains like capture-the-coins).
4) When combining shaping methods, they should be based on different principles to avoid detrimental interactions. For instance, if something changes based on time, something else should change based on a specific behavior.

## VIII. FUTURE WORK

A different environment could provide other dimensions for shaping than the two used in this paper. Given that environmental shaping (in terms of Alternating Rules) turned out so effective in capture-the-coins, it would be interesting to evaluate whether other aspects of the environment could be similarly shaped to maximum effect. For example, a combat-oriented environment could include shapable characteristics in the agents' actions; others might include shapable objects such as walls, roads, and tools. Further, the agents could modify the environment themselves with increasing impact, including the way in which the rules of the environment work during training.

Although automated shaping is more efficient in the current domain, human-driven shaping may be more powerful in other ways. For instance, human creativity could allow new and interesting behaviors to emerge from coevolution that otherwise would not. The challenge will be to anticipate such creativity by providing sufficient means to the human designer to change the problem.

The limits of shaping could be explored by allowing the shaping parameters to more drastically change the environment: Is there some point where such changes become counterproductive, and likewise, is there a point where shaping is most effective? The populations produced over long-term shaped coevolution could be analyzed, identifying the traits that successfully shaped and coevolved populations have in common, perhaps making it possible to determine which design problems are most amenable to the technique.

## IX. CONCLUSION

Prior research in both shaping and coevolution has demonstrated the power of each technique individually. While they are both addressing the same problem in evolution, i.e. continual progress from easier to more challenging tasks, they are in principle compatible and can be combined to an improved effect. Several methods of automatic shaping of coevolution were introduced and compared in this paper. Each of them is better than not shaping, and successes and failures of each shaping method lead to a general understanding of what works and why. In particular, shaping works through increasing diversity in an informed way, and shaping the environment is more effective in this process than shaping the fitness, at least in the capture-the-coins task. In the future, this understanding can be used to design an automated shaping method for a given domain, thus allowing us to create more intelligent game agents than was possible before.

## REFERENCES

[1] P. Darwen and X. Yao. Automatic modularization by speciation. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC '96)*, pages 88–93, Piscataway, NJ, 1996. IEEE Computer Society Press.

[2] R. Dawkins and J. R. Krebs. Arms races between and within species. *Proceedings of the Royal Society of London Series B*, 205:489–511, 1979.

[3] E. De Jong and J. B. Pollack. Ideal evaluation from coevolution. *Evolutionary Computation*, 12:159–192, 2004.

[4] J. L. Elman. Incremental learning, or The importance of starting small. In K. J. Hammond and D. Gentner, editors, *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, pages 443–448. Hillsdale, NJ: Erlbaum, 1991.

[5] S G. Ficici and J. B. Pollack. Pareto optimality in coevolutionary learning. In J. Kelemen, editor, *Sixth

*European Conference on Artificial Life*. Berlin: Springer, 2001.

[6] F. Gomez and R. Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342, 1997.

[7] I. Karpov, J. Sheblak, and R. Miikkulainen. OpenNERO: A game platform for ai research and education. In *Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2008.

[8] W. B. Knox and P. Stone. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*, 2010.

[9] J. B. Pollack, A. D. Blair, and M. Land. Coevolution of a backgammon player. In C. G. Langton and K. Shimohara, editors, *Proceedings of the 5th International Workshop on Artificial Life: Synthesis and Simulation of Living Systems (ALIFE-96)*. Cambridge, MA: MIT Press, 1996.

[10] C. D. Rosin and R. K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5: 1–29, 1997.

[11] C. D. Rosin. *Coevolutionary Search Among Adversaries*. PhD thesis, University of California, San Diego, San Diego, CA, 1997.

[12] T. Savage. Shaping: The link between rats and robots. *Connection Science*, 10:321–340, 1998.

[13] J. Schrum and R. Miikkulainen. Evolving agent behavior in multiobjective domains using fitness-based shaping. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2010.

[14] K. O. Stanley, B. D. Bryant, and R. Miikkulainen. Real-time neuroevolution in the NERO video game. *IEEE Transactions on Evolutionary Computation*, 9(6):653–668, 2005.

[15] K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.