

Evolving Strategies for Social Innovation Games

Erkin Bahçeci
Department of
Computer Science
University of Texas at Austin
Austin, TX, 78712, USA
erkin@cs.utexas.edu

Riitta Katila
Department of Management
Science & Engineering
Stanford University
Stanford, CA, 94305, USA
rkatila@stanford.edu

Risto Miikkulainen
Department of
Computer Science
University of Texas at Austin
Austin, TX, 78712, USA
risto@cs.utexas.edu

ABSTRACT

While evolutionary computation is well suited for automatic discovery in engineering, it can also be used to gain insight into how humans and organizations could perform more effectively in competitive problem-solving domains. This paper formalizes human creative problem solving as competitive multi-agent search, and advances the hypothesis that evolutionary computation can be used to discover effective strategies for it. In experiments in a social innovation game (similar to a fantasy sports league), neural networks were first trained to model individual human players. These networks were then used as opponents to evolve better game-play strategies with the NEAT neuroevolution method. Evolved strategies scored significantly higher than the human models by innovating, retaining, and retrieving less and by imitating more, thus providing insight into how performance could be improved in such domains. Evolutionary computation in competitive multi-agent search thus provides a possible framework for understanding and supporting various human creative activities in the future.

Categories and Subject Descriptors

I.2.1 [Artificial Intelligence]: Applications and Expert Systems—*Games*; I.2.6 [Artificial Intelligence]: Learning—*Connectionism and neural nets*

Keywords

Games; neural networks; empirical study; decision making; social science

1. INTRODUCTION

Many human creative activities can be described as competitive multi-agent search (CMAS; [1]), i.e. search where multiple agents search simultaneously for peaks in a common fitness landscape. The agents may observe each other, and the landscape may change as a result of their search. For example,

companies in high-technology industries search for new products, such as smartphones, tablets, and smart watches, taking into account what they already have, what other companies have, and how well they are each doing. Similar search can be seen to take place in science, music, and art, where many people compete for discovering successful solutions in a common field.

Like other search problems such as single-agent search and cooperative multi-agent search, CMAS can be formulated mathematically and simulated computationally [1]. Evolutionary computation is then a possible way to discover effective strategies for it. For instance, simulations in abstract *NK* fitness landscapes showed that in order to maintain high fitness, the agent should hide some of its findings from the competitors (avoiding a “Twitter effect” where everyone follows the same leaders), and the agent should be quick to move to new and emerging areas (exploit the “wave-riding” effect) [1].

While abstract domains are useful in identifying such general principles of CMAS, it is also necessary to verify them in actual human domains. This task is undertaken in this paper, evaluating CMAS in a fantasy league game. Although this game was designed for a cognitive science laboratory study on social learning [9], it turns out appropriate for studying CMAS as well. Data from the laboratory study is used to generate a simulation domain where human search strategies can be characterized, compared, and evaluated wrt. strategies discovered automatically through evolution. More specifically, the first goal is to discover better strategies that people use in this domain, by building models of individual players. The second goal is to evaluate how well those strategies work, and whether it is possible to discover better strategies automatically using evolutionary optimization. For instance, do we need different strategies against opponents that imitate a lot vs. those that innovate more, in order to perform as well as possible? Are there strategies that would perform well in many different environments, or are the best strategies customized to each environment separately?

This paper aims to answer such questions using evolution as the method to optimize neural networks. Three experiments were conducted: First, in order to characterize environments and successful strategies, strategies were evolved against a uniform set of opponents of a certain type. Second, general strategies were evolved using multiple environments for fitness evaluation. Third, in order to show that better strategies can be discovered in complex real-world environments, strategies were evolved in the game groups of the human subject study. The results show that humans do follow a variety of general strategies, and it is possible to do better by opti-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '15, July 11 - 15, 2015, Madrid, Spain

© 2015 ACM. ISBN 978-1-4503-3472-3/15/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2739480.2754790>

mizing the strategy to the actual environment. Thus, these results demonstrate that CMAS is a productive formulation of a new area of human problem solving, making it amenable to analysis and optimization.

The paper is organized as follows: In Section 2, CMAS is first motivated from the real-world perspective of innovation search. A formalization of CMAS is given, and basic computational results on an abstract NK landscape reviewed. The social innovation game domain is described in Section 3, and how human subjects were modeled in this domain is described in Section 4. Section 5 describes how strategies were encoded and evolved in this domain, and presents a series of experiments comparing evolved strategies against human models.

2. BACKGROUND

The original motivation for CMAS comes from innovation search, which is part of organizational theory in management science. In this theory, search (i.e. organizational problem solving) is thought to take place in a *knowledge space*, conceptualized as a landscape. In particular in innovation search, firms generate, recombine, and manipulate knowledge within a pool of technological possibilities [3, 6]. The agents do not know the topography of the landscape but can sample it, i.e. search the landscape by generating and evaluating new points.

More formally, the search agent’s knowledge $\mathcal{X}(t)$ of the landscape and its topography at time t consists of m points with fitness values z :

$$\mathcal{X}(t) = \{[\mathbf{x}_1, z(\mathbf{x}_1)], [\mathbf{x}_2, z(\mathbf{x}_2)], \dots, [\mathbf{x}_m, z(\mathbf{x}_m)]\}. \quad (1)$$

In the traditional *single-agent search*, the agent moves to the next (i.e. $(m + 1)$ th) point using a search strategy S based on what the agent already knows about the landscape, i.e. $\mathbf{x}_{m+1} = S[\mathcal{X}(t)]$. For instance, in order to find the best of a kind, the agent might search near the good points that it has already discovered (*exploit*), or if the existing points do not seem promising, avoid them, and make a long jump to an area far from them (*explore*). The search strategy S is stochastic, i.e. it contains an element of uncertainty due to bounded rationality. A good search strategy is a method for visiting new points such that high peaks are found as fast as possible.

In contrast, in *competitive multi-agent search* (CMAS; Figure 1), this formulation is extended in two ways: (1) The agent’s knowledge of points \mathcal{X} includes not only the points \mathcal{X}_h it has discovered itself and keeps hidden from other agents, but also points \mathcal{X}_p that the other agents have found and made public; (2) whenever new points are found in the neighborhood ρ of existing points, all their values are changed with a multiplicative factor α . These extensions model the knowledge that the agents have about their competitors’ search, and the dynamic effects that the competitors’ searches have on the landscape. Thus, a competitive search strategy consists of two components:

$$\mathbf{x}_{m+1} = S_1[\mathcal{X}_h(t), \mathcal{X}_p(t), \rho(t), \alpha(t)], \quad (2)$$

$$\begin{aligned} \mathcal{X}(t + 1) &= \{\mathcal{X}_h(t + 1), \mathcal{X}_p(t + 1)\} \\ &= S_2[\mathcal{X}_h(t), \mathcal{X}_p(t), \mathbf{x}_{m+1}, z(\mathbf{x}_{m+1}), \rho(t), \alpha(t)], \end{aligned} \quad (3)$$

where S_1 generates a new point \mathbf{x}_{m+1} given the knowledge about points and about the dynamic effects at time t , and S_2 updates this knowledge with the new point, i.e. generates the version of $\mathcal{X} = \{\mathcal{X}_h, \mathcal{X}_p\}$ for time $t + 1$, where one of those

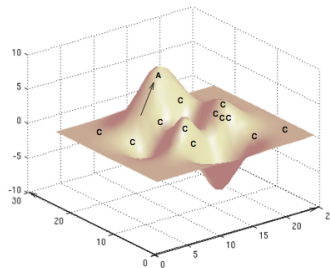


Figure 1: A conceptual illustration of CMAS search. Multiple agents search for fitness peaks in the same environment, modeling human creative problem solving.

knowledge bases includes the new point \mathbf{x}_{m+1} and its fitness value $z(\mathbf{x}_{m+1})$.

This formulation of CMAS makes a new class of human problem-solving activity amenable to formal and computational analysis. The first step is to understand the general properties of CMAS problems. A good approach is to analyze CMAS in an abstract, mathematical domain that makes the conclusions clear. The binary NK landscape [4], where N is the dimensionality and K determines how complex the relationships between the dimensions are, serves as such a domain.

In a previous study of CMAS in an NK landscape [1], S_1 was represented as a vector of probabilities for taking an exploit and explore action from the best private and public point seen so far, based on the fitnesses of those points, and S_2 was represented as a vector of probabilities of placing the new point in private or public memory, based on the fitness of that point. Different search environments were established by creating opponent agents with different preferences for these actions. Such preferences determine how the agent searches (i.e. their strategy), which can be optimized through evolution. In prior work, specialized strategies were evolved for an agent for each different environment, and also a general strategy to perform well across multiple environments. These strategies turned out more effective and more complex than hand-designed strategies and a strategy based on traditional single-agent tree search. Using a novel spherical visualization of NK landscapes, insight was gained about how successful strategies work, including the Twitter and wave-riding effects mentioned in Section 1.

The next step, to be undertaken in this paper, is to evaluate these conclusions in a domain with actual human subjects. Such a domain will be described next.

3. THE SOCIAL INNOVATION GAME DOMAIN

To study competitive multi-agent search in a human domain, a dataset of human behavior in a competitive multi-agent search task was employed: a social innovation game created by Wisdom et al. [9]. The human dataset was collected under laboratory conditions at the Percepts and Concepts Laboratory of Robert L. Goldstone at Indiana University¹. The task that human subjects performed was a multi-player problem-solving game similar to a fantasy sports league: The players tried to build a collection (or team) of icons that would score higher than the team of other players. The subjects, who were undergraduate students, were assigned to groups of size one through nine, in which they each played eight games.

A screenshot of the game’s graphical user interface can be seen in Figure 2. During games that consisted of 24 rounds, each player built a team of five or six members (shown as *icons*

¹<http://cognitntrn.psych.indiana.edu/>

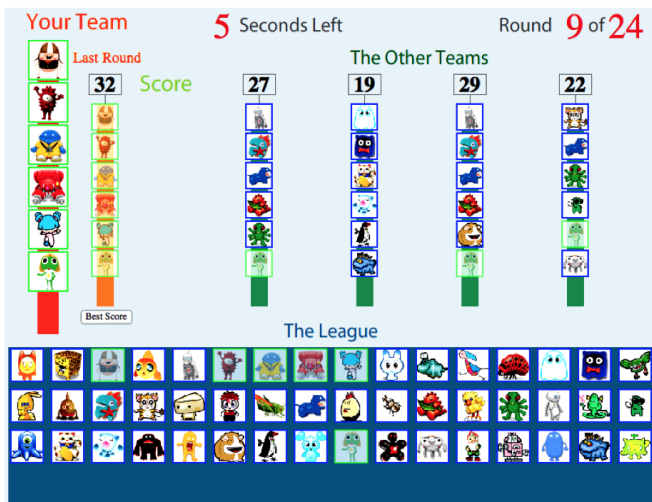


Figure 2: The graphical user interface for the social innovation game [9]. The player’s current team is shown at the left hand side (labeled “Your Team”) and the opponents teams at top right (labeled “The Other Teams”). The objective is to build a team of icons that has a better score than the other players’ teams. The user interface also displays the current round and the total number of rounds at the top right corner, and the remaining time in the current round at the top center. (Used with permission from Robert L. Goldstone.)

on the interface) in each round. The players could add icons to their teams by dragging an icon from a source, and dropping it onto one of the player’s own icons, replacing it. Icons could be copied in this manner from four sources: (1) player’s last team (a *retaining* action), (2) player’s best scoring team up until that point in the game (a *retrieval* action), (3) another player’s last team (an *imitation* action), or (4) the league of all available icons (an *innovation* action). Depending on the game configuration, there were either 24 or 48 available icons in the league. The players could also copy any of the other teams in its entirety by dragging the score label above the team, and dropping onto their own current team. In each game, a fixed number of points was assigned to each icon in the league, and bonuses and penalties were assigned to a subset of distinct pairs of icons. These point assignments, which were not known by the human subjects, were determined by Wisdom et al. to make the game challenging by giving high bonuses to least valuable icons and high penalties to most valuable icons. At the end of each round, each player’s score was calculated as the sum of points assigned to the individual icons in the player’s team, as well as any bonuses or penalties for icon pairs. In each game, a fixed number of points in the range [1, 8] were assigned to each icon in the league, and penalties or bonuses in the range [-20, 20] were assigned to a subset of distinct pairs of icons. The range of possible scores was [-6, 60], and actual scores of human players were in the range [-2, 60].

The dynamics of this domain are simpler than those studied with abstract simulation on NK fitness landscapes [1] in that the fitness landscape does not change during the game. However, the domain is still a CMAS, and serves as a real-world example and application of competitive multi-agent search. Moreover, the human dataset allows modeling people’s strategies in such domains, as well as determining if there are

better strategies, through the use of evolutionary computation.

4. MODELING HUMAN SUBJECTS

Among other effects, Wisdom et al. identified effects of game round on the subjects’ behavior in the aforementioned social innovation game [9]. The proportion of actions (i.e. icon sources) that the players employed changed with the round. As the game progressed, retention and retrieval increased slightly, while imitation and innovation decreased but overall the proportion of actions remained relatively stable. On average, subjects imitated 9.8%, innovated 13.7%, retained 73.9%, and retrieved 2.6% of the time. In sum, subjects were in general conservative in their gameplay, but became even more so over the course of 24 rounds.

Since those action ratios are known for human subjects, they can be used as distance measures comparing a model’s behavior to that of a particular human subject. The smaller the difference between action ratios, the more useful a model is as a replacement for a human subject in a simulation. Thus, those action ratios, as well as the score of the players and the consistency of their icon choices, were used as distance objectives to evaluate models in this paper.

In order to simulate human subjects, gameplay data from each human subject was used to train a model for the subject in the form of two separate neural networks: one for high-level actions (i.e. copying a whole team) and another one for low-level actions (i.e. copying individual icons). Both types of neural network models chose actions probabilistically based on the relative activation of their output units, and were trained with backpropagation using 500 epochs, using a learning rate of 0.1, momentum of 0.1, and weight decay of 0.01.

The models with two separate neural networks were compared to the corresponding models created with two other modeling approaches based on human gameplay data, as well as a baseline model with uniform action choice. More specifically, four models were thus compared in Figure 3: (1) the baseline model where icon actions are picked based on fixed and uniform probabilities (i.e. 0.25 for each of the four icon actions), (2) another simplistic model with fixed probabilities, but with each action’s probability set as the human subject’s usage ratio for that action, calculated from the data, (3) a one-tiered neural network model that performs icon actions but not team actions, and (4) the two-tiered neural network model, which performs both team and icon actions.

While there was no significant difference between the two neural networks in innovation ratio, icon consistency, or the two score objectives, they did differ in three other distance objectives, including team and icon imitation ratio (i.e. first and second on the second row in Figure 3). The two-tiered models were significantly closer to human subjects along these objectives than the other model types (p -value $< 10^{-4}$). The main reason for the difference in team imitation ratio is that the other models did not perform any team actions, and therefore got the same result on team action ratio objectives (i.e. “team imitation ratio” and “team retrieval ratio”). However, unlike in the team imitation ratio, two-tiered models did not perform significantly better in the team retrieval ratio. A likely reason is that human subjects rarely performed that action: 27 out of 34 subjects did not do team retrieval at all, while the others did very rarely. The two-tiered models learned to rarely perform that action, which was not significantly different from not doing it at all. While the two-tiered models were

Distance to Human Subjects

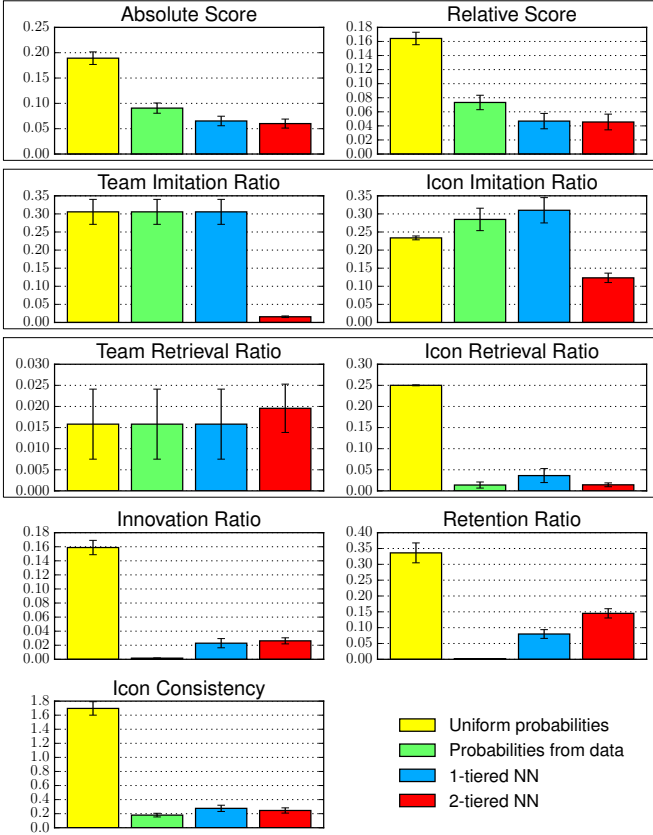


Figure 3: Different modeling approaches compared using several distance objectives. Values show average distance of the models to the corresponding human subject. Smaller values are better. Each distance objective is averaged across the human subjects. Error bars indicate one unit of standard error of the mean. Note that the y -axes have different scales. Related objectives are grouped in surrounding boxes. The two-tiered neural network models do significantly better than all others in team and icon imitation ratio objectives, whereas the difference with the one-tiered networks are not significant in other objectives. The reason why two-tiered network models do not do better in retrieval ratio objectives is that retrieval is done very rarely by humans, and therefore insufficient data for team and icon retrieval actions. The simpler models (i.e. second bar) that are based on probabilities from human gameplay data perform better in innovation and retention ratio objectives, but they cannot perform team and icon actions separately, unlike the two-tiered network models.

significantly worse in the retention ratio (p -value < 0.01), the advantage of two-tiered network models in the two imitation action ratio objectives and in their support for team actions make the two-tiered network models a better choice for modeling human subjects than the one-tiered ones.

The fact that these human models come close to human subjects in terms of their action ratios confirms that learning to imitate human players results in models that can be characterized like Wisdom et al. did as mentioned above. Besides characterizing human behavior, these models also allow simulating actual human players. It is much more practical

to create hypothetical situations using models of human subjects than doing the same with the subjects themselves, which is important in optimizing strategies, as will be done in the next section.

5. OPTIMIZING SOCIAL INNOVATION

With models that mimic human players, one can create simulations of game environments that resemble those with human players. Such environments can then be used to discover strategies that perform well against human players, and ones that possibly perform even better than humans. An effective way to do so is through evolutionary computation. For instance, player strategies, encoded as neural networks, can be evolved by evaluating their fitness in game environments with human model opponents. A strategy encoding that is suitable for evolution will be described next.

5.1 Two-tiered Combined Neural Network

In Section 4, two separate neural networks were used to model human subjects. However, if such a model was used for evolving strategies, hidden network nodes that represent useful states would have to evolve separately twice (i.e. once per neural network). For instance, a hidden node may represent the state where the game is about to end and the relative score of the agent is lower than the player with the highest score, and the output of this hidden node may be useful for both high-level and low-level actions. Sharing such hidden nodes between the two network tiers would allow neural network representations to be evolved once and used for both high-level and low-level actions. Therefore, employing a combined neural network can reduce the search time for a neural network solution.

Thus, for the goal of optimizing strategies, a combined two-tiered neural network architecture was chosen to represent strategies. The neural network nodes had either a sigmoid or a Gaussian function as their activation function. As seen in Figure 4, high-level (i.e. team-level) actions and low-level (i.e. icon-level) actions were generated by separate outputs of the same neural network, as opposed to outputs of two separate neural networks as in Section 4. The next section details how these networks were optimized for the social innovation game domain.

5.2 Experimental Setup

The strategy for a single agent was optimized given an environment with a fixed set of opponents. As described in the previous section, a single neural network was used to represent a strategy, which made it straightforward to use neuroevolution methods to optimize it. Since NEAT [8] is a commonly used and effective neuroevolution method, it was used in this paper for evolving strategies.

Strategies were evolved in environments with eight opponents, each of which was the model of a human subject, created using supervised learning as described in Section 4. NEAT ran for 500 generations for each environment, with a population of size 100, and with 64 repetitions for each evolutionary setup.

The fitness of each evolved strategy was evaluated by running a simulation where the first agent used the evolved strategy and the opponents used a fixed strategy chosen to create a particular environment. The fitness was calculated as the normalized score of the first agent averaged across 200 games. As in human subjects' games, at the beginning of each game, each player had a team that was randomly selected from the

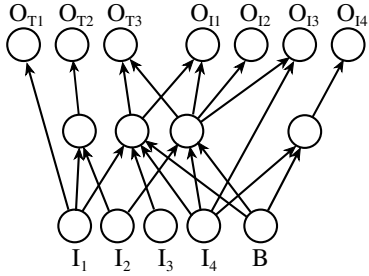


Figure 4: An example combined two-tiered neural network, with four inputs (i.e. current game round, player’s score relative to the best opponent, icon’s popularity, icon’s age), three outputs for team actions (i.e. imitation, retrieval, and no action), and four outputs for icon actions (i.e. imitation, innovation, retrieval, and retention or no action). The outputs for icon actions are normalized and used as action probabilities for each icon of the player’s team, while team action outputs are normalized separately and used as team action probabilities. In this design, the team and icon networks are combined, as opposed to the separate networks in Section 4. Hidden nodes representing game states that are useful for both team and icon actions are shared, and do not have to be evolved twice.

league. Moreover, as in human games, icon memorization was effectively prevented, since agents did not guess or keep scores for individual icons or pairs of icons.

With this setup, three experiments were conducted. In the first experiment, custom strategies are evolved for environments where all the opponents used the same strategy (Section 5.3). The second experiment investigates whether general strategies that would work across multiple such environments could be evolved (Section 5.4). These experiments demonstrate what is possible by evolving strategies for extremely different environments. The third experiment evaluates these questions in more realistic environments where the opponents use different strategies, modeling the actual games played in the laboratory study (Section 5.5).

5.3 Experiment 1: Evolving Specific Strategies

In order to see which strategies work well in different social innovation game environments, several evolutionary setups were created, each with a different set of opponent models. For each of the four icon actions (i.e. imitate, innovate, retain, and retrieve), the model for the subject that used that action the most was selected as the dominant model for that action, among the models for subjects who took part in games with at least five players. For each action, eight copies of the corresponding action-dominant model were used as the opponents in an environment, resulting in four homogeneous environments.

To make it easier to examine evolved strategies, the corresponding networks were used as CPPNs to generate patterns to fill state-action probability tables. The possible range of input values for the four CPPN inputs were discretized into three coarse values for the time input, and two for the remaining inputs: (1) beginning of game, mid-game, or end-game; (2) low or high score relative to the best opponent; (3) low or high icon popularity among players; and (4) old or new icon. Combinations of these input values correspond to 24 possible discrete game states for each icon. For each state, the

agent had a choice of three team actions for the whole team, as well as four icon actions for each icon. As a result, from each evolved CPPN, a 24×7 state-action probability table was obtained.

To identify the differences between strategies evolved in different environments, action probabilities were compared using Student’s t-test, separately for each action and for each of the 24 discrete states. This comparison resulted in 24 p -values for each pair of strategies median p -value was then used to determine whether strategies evolved in an environment had significantly higher or lower probabilities than those evolved in another environment.

Strategies evolved in the environment with imitate-dominant opponents do significantly more innovation (with median p -value $< 10^{-3}$) and less imitation (with median p -value for icon and team imitation 0.0134 and 0.0454, respectively), as well as more retrieval (with median p -value 0.0287). In such an environment, the source of imitation is always the highest-scoring player. Therefore, lower imitation and higher innovation and retrieval lead to higher diversity, which increases the chance of finding icons with higher score, beating the opponents.

On the other hand, strategies evolved with innovation-dominant opponents had significantly higher imitation at the team level (median p -value = 0.022) compared to strategies from other homogeneous environments, while there was no significant difference in icon imitation. A possible explanation is that when all other teams are innovating a lot (i.e. replacing their icons with random ones from the pool), team imitation is more reliable than icon imitation. Strategies evolved against retention-dominant opponents have increased team imitation as well, but it is not as significant (median p -value = 0.066).

With retrieve-dominant opponents, evolved strategies innovated significantly less (median p -value $< 10^{-4}$) than in other environments, and imitated somewhat more at the icon level (median p -value = 0.094). Innovation is getting a completely random icon, whereas icon imitation copies a random icon from the best-scoring opponent’s team, which, in this environment, opponents are likely to have icons from the opponent’s best team so far. Therefore, in this environment innovation is less likely to contribute positively to a player’s score than imitation.

The 24×7 action probability tables were also used to examine the distribution of strategies. The probability tables of evolved strategies were treated as 168-dimensional vectors, and reduced to two-dimensional points via principal component analysis (PCA). The first two components captured 83% of the variance in the high-dimensional vectors, and therefore two-dimensional PCA plots were used to visualize the strategies. Table 1 shows the distribution of the best strategies from the 64 evolutionary runs for each environment.

The diagonal boundary line at the bottom left of the PCA plots represents maximum icon imitation (i.e. with probability 1.0), and the one at the bottom right corresponds to maximum team imitation; therefore there are no points below those two lines. The result that the strategies evolved against imitation-dominant opponents imitated less means that the points (i.e. strategies) are further away from the diagonals than in the other environments. The relatively high team imitation evolved against innovation- and retention-dominant opponents can also be observed in these plots: there is a strong cluster around the diagonal line at the right representing maximum team imitation in these environments. Similarly, for strategies evolved against retrieve-dominant opponents, rela-

Environment		PCA for Evolved Strategies
Id	Opponents	
1	Imitate-dominant (8)	
2	Innovate-dominant (8)	
3	Retain-dominant (8)	
4	Retrieve-dominant (8)	
5	Heterogeneous (2 opponents from each of Environments 1-4 above)	
6	Multiple homogeneous (Fitness averaged over Environments 1-4 above)	

Table 1: Strategies evolved in homogenous and heterogeneous environments. Each PCA plot shows 64 evolved strategies. The PCA dimensions are shared across all evolved strategies. The diagonal boundary lines at the bottom left and bottom right (somewhat faint in this table but more clearly visible in Table 2) represent maximum icon imitation and maximum team imitation, respectively.

tively more numerous points can be seen near the diagonal at the left, which corresponds to higher icon imitation.

Performance of the evolved strategies across various environments is shown in Figure 5. For each homogeneous environment, the strategies evolved in the same particular environment significantly outperformed the ones evolved in other environments (p -value $< 10^{-19}$). This result shows that there is indeed value in optimizing the strategies for a particular en-

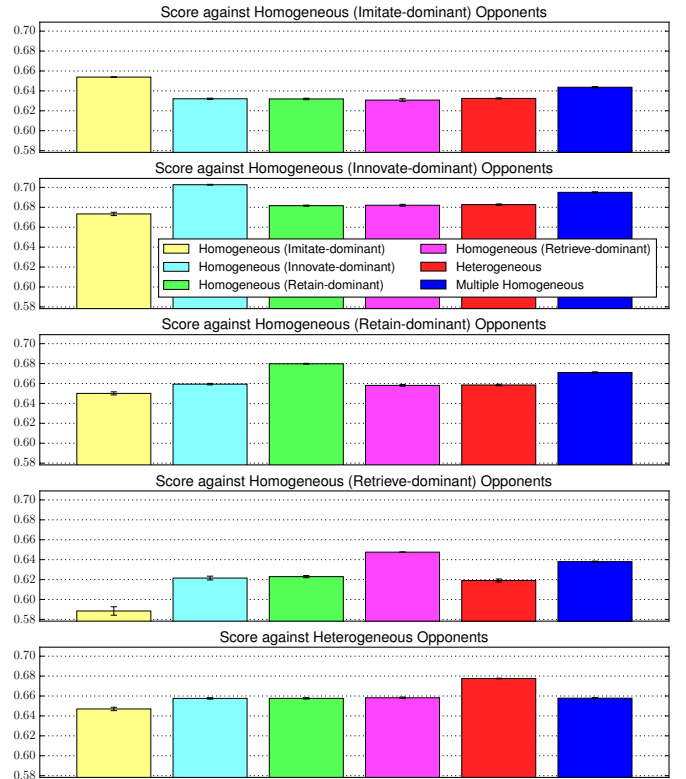


Figure 5: Performance comparison among strategies evolved in a single homogeneous, multiple homogeneous, and a heterogeneous environment. Each row shows the performance of strategies in one environment. Each column shows the average score of 64 strategies evolved in the environment indicated by the legend when they are evaluated in the environments that correspond to rows. Thus, the bars in the diagonal starting from top left show the performance of strategies that were evolved in the same environment as the one in which they are evaluated. Each group of evolved strategies on the diagonal perform significantly better than the other evolved strategies in the same row. Moreover, the strategies evolved in multiple homogeneous environments perform significantly better than others in all homogeneous environments (except the strategies evolved in the evaluation environment). Thus, evolution can produce customized strategies for given homogeneous and heterogeneous environments, as well as general strategies that perform better than others in multiple environments.

vironment. Strategies deployed in foreign environments performed at similar levels with each other, except for the ones evolved against imitate-dominant opponents, which performed significantly worse than the others (p -value $< 10^{-4}$). A possible explanation is that the increased innovation is actually a handicap in those other environments.

While homogeneous environments are useful in evolving strategies to counter a specific type of opponent, a different approach is needed for evolving general-purpose strategies that can be used against different opponents, which will be the focus of the next section.

5.4 Experiment 2: Evolving General Strategies

In order to evolve general strategies, two more evolutionary setups were employed: (1) a setup where fitness of strategies

are evaluated by averaging the score of the strategy across all four homogeneous environments, and (2) a setup with a heterogeneous environment where two copies of each of those four action-dominant models were used as opponents.

Figure 5 shows that strategies evolved in heterogeneous environments also performed better than other strategies in the heterogeneous environment itself, but did not show any benefit over other strategies in homogeneous environments. On the other hand, while strategies evolved in multiple homogeneous environments performed worse in each homogeneous environment than the ones evolved in that particular environment, they significantly outperformed the rest of the strategies (p -value $< 10^{-11}$). These results suggest that evolving strategies in a set of diverse homogeneous environments may be a useful approach to create strategies that generalize well.

So far, evolutionary environments consisted of an artificial combination of models, in order to understand how the different strategies interact. The next section looks at evolutionary results with sets of opponents that are determined in a more realistic way, taking into consideration the groups in the human subject experiment data.

5.5 Experiment 3: Evolving in Complex Environments

To see how strategies evolve in environments that are more realistic than the ones employed so far, evolutionary environments were created to simulate three groups of eight or nine players from the human experiment data. For each player in each group, the player was replaced with the evolving strategy and the opponents with the corresponding human subject’s models that were created in Section 4, resulting in nine evolutionary environments for Group 1, and eight for Groups 2 and 3, with a total of 25 environments. Each pair of environments within each group shared all but one opponent, which makes them more similar compared to environments from the two other groups.

As in the previous experiments, 64 evolutionary runs were performed for each environment, and the best strategy at the end of each run was selected as the resulting evolved strategy from that run. Table 2 shows the distribution of those strategies, organized by originating human group. Since there are many more strategies displayed in the PCA plots, the two diagonal boundary lines mentioned above at the left and right are much more prominent in Table 2 than in Table 1.

Furthermore, a few more lines parallel to those two are visible about half-way and one-third of the way from the bottom convergence point to the left and right edges. Just like the left and right boundary diagonals correspond to maximum icon and team imitation, respectively, these lines correspond to 50% or 67% icon or team imitation. This result is likely an artifact of the employed configuration of NEAT, where each member of the initial evolution population is a minimal network, often having a zero value for all except one output of the network, which gets interpreted as a fractional action probability such as 0.5, 0.33, or 0.67. With more strategies in the PCA plots, it is also clearer that the corresponding points lie in a triangle (i.e. the bottom half of a rotated square), which means most of them have at least 50% of team or icon imitation.

The performance of the evolved strategies, grouped by originating human groups can be seen in Figure 6. As in Experiments 1 and 2, strategies evolved with similar opponents (i.e. within each group) performed significantly better in environ-

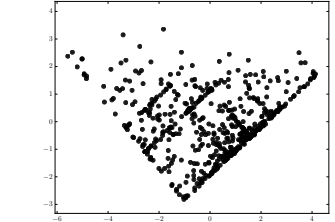
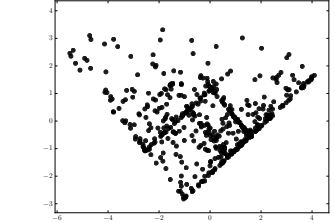
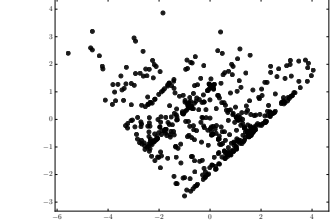
Environment		PCA for Evolved Strategies
Id	Opponents	
1	Group 1 (8) × 9	
2	Group 2 (7) × 8	
3	Group 3 (7) × 8	

Table 2: Strategies evolved in environments with model groups that correspond to three human subject groups. For each group, the same number of environments were created as the number of players in the group, where in each environment, one model was replaced with the evolved strategy during fitness evaluation. The first PCA plot shows $64 \times 9 = 576$ evolved strategies since there are nine players in the first group, whereas the second and third plots show $64 \times 8 = 512$ strategies since those groups had eight players. The PCA dimensions are shared with the evolved strategies in Table 1. The diagonal boundary lines at the bottom left and bottom right represent maximum icon imitation and maximum team imitation, respectively, and they are more prominent than in Table 1 here due to the increased number of displayed strategies.

ments from the same group compared to strategies evolved in the other two groups’ environments (p -value $< 10^{-32}$). The performance difference was smaller than in the first two experiments, which is to be expected because the environments are more similar in Experiment 3 than in Experiments 1 and 2. It is also more difficult to characterize the strategy difference upon which these performance improvements are based. They are subtle and numerous, and in combination, allow the strategy to beat its opponents.

Most interestingly, the evolved strategies significantly outperformed the human subject models that they replaced during evolution by a difference of 0.1 normalized score (p -value $< 10^{-5}$). The score advantage of the evolved strategies can be explained by the difference in action ratios. Overall, the evolved strategies imitated 54% more than the human models, with 52% more team imitation (p -value $< 10^{-10}$). On the other hand, the evolved strategies retained their icons 41% less (p -value $< 10^{-8}$), retrieved icons 4.4% less (p -value = 0.044), and innovated 4% less than the models (p -value =

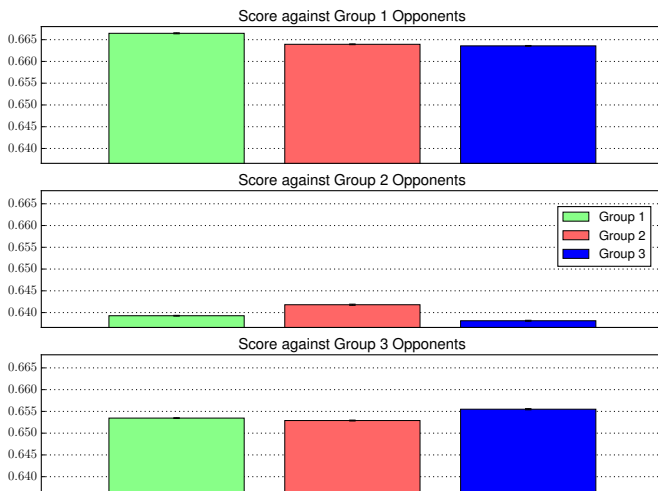


Figure 6: Performance comparison among strategies evolved in environments that correspond to three human subject groups, which are also compared with the average performance of models for human subjects in the group. The layout is similar to Figure 5. Again, the groups of strategies on the diagonal perform significantly better than the others shown in the same row (i.e. those evolved in other environments), though the difference is smaller than in Figure 5. The results demonstrate that evolution can produce customized strategies for given environments in this domain.

0.014). This result shows that it is indeed possible for evolution to discover and utilize opportunities in realistic human competitive multi-agent environments, and improve behavior over humans. Moreover, this result also provides insights into how human performance could be improved.

6. FUTURE WORK

The results in this paper suggest a possible application for CMAS: Creating competitive agents for social innovation games, such as fantasy sports leagues. Further, using the same approach, it should be possible to model and analyze other human problem solving activities like innovation search.

For general strategies like those evolved in Section 5.4 to scale up to even more diverse environments, it might be beneficial to allow multimodal behaviors, so that the same agent strategy can have distinct behaviors in different contexts [7]. A further step in this direction would be to coevolve all or a subset of the opponents as well. In this manner, the environment could present more diverse challenges, and more interesting and perhaps realistic general strategies could evolve.

A limitation of the experiment in Section 5.5 was that strategies were optimized against opponent models for only three groups of human subjects. More subject groups can be used as opponents; such an extension would make it possible to investigate the differences between strategies optimized against different sized groups. Because it is already known that group size affects ratios of utilized actions and the score, such a comparison might yield insights into what strategies work best against such diverse groups.

One downside of the evolutionary approach utilized in this paper is that it requires a large population and therefore a significant number of fitness evaluations. Thus, alternative evolutionary methods such as CMA-ES [2] and Estimation of Distribution Algorithms [5] may be considered. In those

approaches, statistical models of the solutions and their fitness are maintained, making it possible to get by with fewer fitness evaluations in structured domains like CMAS.

7. CONCLUSION

This paper focused on optimization of strategies in the social innovation game domain. Three experiments explored strategies evolved in various environments with opponents that are models of subjects from the human study: (1) single homogeneous environments with opponents that dominantly perform one type of action, (2) multiple homogeneous environments and a single heterogeneous environment, and (3) environments with sets of opponents representing three diverse subject groups from the human study. The second experiment demonstrated that strategies that generalize across diverse sets of opponents can indeed be evolved. However, the first and third experiments both demonstrated that evolution was able to produce stronger strategies by tailoring them to the particular environment. The third experiment in particular showed that it is possible to discover and utilize subtle opportunities in realistic environments, and perform better than the human models by imitating more, and retaining, retrieving, and innovating less. These conclusions suggest that CMAS is a productive approach to understanding and perhaps even automating discovery in similar human creative problem solving domains.

8. ACKNOWLEDGMENTS

This research was supported in part by NSF grants DBI-0939454, IIS-0915038, and SBE-0914796, and by NIH grant R01-GM105042. The authors would like to thank Thomas N. Wisdom and Robert L. Goldstone at Indiana University for providing data and support for the game domain.

9. REFERENCES

- [1] E. Bahceci. *Competitive Multi-Agent Search*. PhD thesis, The University of Texas at Austin, Austin, TX, 2014.
- [2] N. Hansen, S. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.
- [3] R. Katila. New product search over time: Past ideas in their prime? *Academy of Management Journal*, 45(5):995–1010, 2002.
- [4] S. A. Kauffman. *The Origins of Order*. Oxford University Press, New York, 1993.
- [5] P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, volume 2. Springer, 2002.
- [6] D. A. Levinthal and J. G. March. A model of adaptive organizational search. *Journal of Economic Behavior and Organization*, 2:307–333, 1981.
- [7] J. Schrum and R. Miikkulainen. Evolving multimodal networks for multitask games. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2):94–111, June 2012.
- [8] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.
- [9] T. N. Wisdom, X. Song, and R. L. Goldstone. Social learning strategies in networked groups. *Cognitive Science*, 37:1383–1425, 2013.