Artif Life Robotics (2006) 10:157–161
DOI 10.1007/s10015-005-0354-8

ORIGINAL ARTICLE

Yasuyuki Suzuki · Takaya Arita

A comprehensive evaluation of the methods for evolving a cooperative team

Received and accepted: November 30, 2005

Abstract This article focuses on the techniques of evolutionary computation for generating players performing tasks cooperatively. However, in using evolutionary computation for generating players performing tasks cooperatively, one faces fundamental and difficult decisions, including the one regarding the so-called credit assignment problem. We believe that there are some correlations among design decisions, and therefore a comprehensive evaluation of them is essential. We first list three fundamental decisions and possible options in each decision in designing methods for evolving a cooperative team. We find that there are 18 typical combinations available. Then we describe the ultimately simplified soccer game played on a one-dimensional field as a testbed for a comprehensive evaluation for these 18 candidate methods. It has been shown that some methods perform well, while there are complex correlations among design decisions. Also, further analysis has shown that cooperative behavior can be evolved, and is a necessary requirement for the teams to perform well even in such a simple game.

Key words Genetic algorithm · Cooperative behavior · Multiagent system · Ultimately simplified soccer game

1 Introduction

Some problems can be solved efficiently only by teams consisting of cooperative autonomous players. Many researchers have developed methods that do not require human designers to define the specific behaviors of players for each problem. The work reported here focuses on the techniques of evolutionary computation, which has been regarded as one of the most promising approaches to solving such complex problems. However, in using evolutionary computation for generating players performing tasks cooperatively, one faces fundamental and difficult decisions, including the one regarding the so-called credit assignment problem.[1] For example, if we can only evaluate the global performance of each team, how do we divide up the team's performance among the participating players? We believe that there are some correlations among design decisions, and therefore a comprehensive evaluation of them is essential, although several researchers have proposed evolutionary methods for evolving teams performing specific tasks.

The rest of this artical is organized as follows. In Sect. 2, we list three fundamental decisions and possible options in each decision in designing a method for evolving a cooperative team. We find that there are 18 typical combinations available. Then in Sect. 3, we describe the ultimately simplified soccer game played on a one-dimensional field as a testbed for a comprehensive evaluation of these 18 candidate methods. Sect. 4 reports on the results of the comprehensive evaluation of these methods, and Sect. 5 summarizes the work.

2 Methods for evolving a team

Three fundamental decisions are necessary when one designs an evolutionary computation method for generating players performing tasks cooperatively, and there may be several combinations of options in these decisions.

The first decision is: How many evolving populations are there? The answer is derived by considering whether or not the population structure depends on the number of teams in the game, or on the number of player roles in the game (Fig. 1). Suppose that the game is played by 2 teams each consisting of 3 players. We can assume an evolutionary computation with 2 populations corresponding to 2 teams, with 3 populations corresponding to 3 players, or with 6

Y. Suzuki · T. Arita (✉)
Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
Tel. +81-52-789-3503; Fax +81-52-789-3503
e-mail: arita@nagoya-u.jp

This work was presented in part at the 10th International Symposium on Artificial Life and Robotics, Oita, Japan, February 4–6, 2005

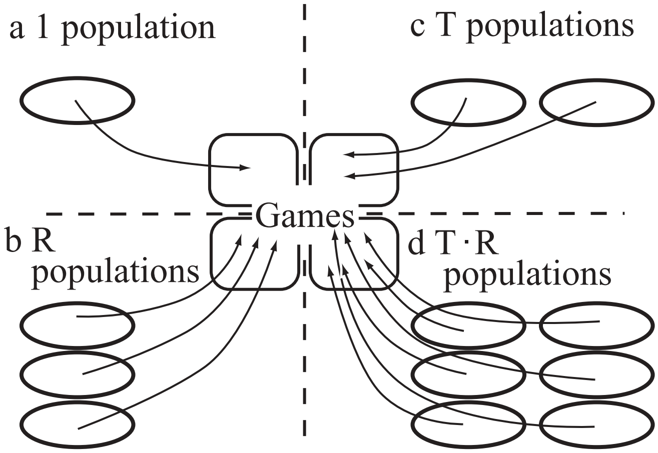

Fig. 1a–d. The four options for the population structure. **a** The population represents all player roles in all teams. **b** Each population represents one player role in all teams. **c** Each population represents all player roles in each team. **d** Each population represents one player role in each team

populations corresponding to 2 teams of 3 players. So the typical options for the number of populations are 1, R, T, and $T \cdot R$ (T, number of teams in the game; R, number of player roles in the team).

The second decision is: What does each individual (genome) represent? Typical options are a player and a team. In the case where each genome represents a player, there can be two further options: all players in the team share one genome ("homogeneous players"), or all players are represented by different genomes ("heterogeneous players"). In the case where each genome represents a team, there can be two further options: whether or not the roles of the players represented in each genome are fixed. In the case where the roles of the players are fixed, for example, if a part of a genome represents a defender in the game, this part always represents a defender.

The third decision is: How is the fitness function evaluated? One option is that fitness is evaluated for a team as a whole. In this case, if each genome represents a player, each player in a team is supposed to have the same fitness. The other option is that the fitness is evaluated for each player directly or indirectly. Direct evaluation of players in a cooperative team is sometimes a very difficult task, as in general altruistic behavior is important or essential in the establishment and maintenance of cooperation in a population. Some methods for indirect evaluation have been proposed.[2] We adopt a method in which the fitness of a player is defined as the decrease in the fitness of the team when the player is replaced by a predefined "primitive player" who has a minimum set of behavior rules.

Therefore, there could be 18 available combinations for evolving players performing tasks cooperatively, as shown in Table 1.

Many researchers have treated this issue, although most of them focused on one or two methods of the 18 combinations. Some significant studies are now described. 1–PHo–T is the simplest method, in which there is one population and all players in a team share one genome. Quinn et al.[3] adopted this method, and successfully evolved agents that specialize, based on their relative positions, in order to perform better as a team, although the agents were homogenous. Miconi[2] adopted 1–PHe–PI, in which the fitness of each individual was determined as the decrease in fitness when that individual was not present in the team in the context of on-line evolution. Luke[4] evolved teams of soccer players through an adapted version of genetic programming: homogenous teams (1–PHo–T) and heterogeneous teams (1–TFi–T). Potter and DeJong[5] proposed cooperative evolutionary algorithms, which can be classified into R–PHe–T, and tested them in the domain of function optimization, in which each population contained values for one parameter, and the fitness of a parameter was obtained by combining the current best parameters of the remaining populations. Our previous study[6] compared [1–PHe–PD], [1–PHe–T], [1–TFi–T], [R–PHe–PD], and [R–PHe–T] using a multirobot model in which not only control of behaviors, but also morphology (including selection and the arrangement of sensors/motors), evolved via ontogenesis.

3 Ultimately simplified soccer game

The ultimately simplified soccer game is defined as a testbed for a comprehensive evaluation for these 18 candidate methods. It is a 2 vs 2 player game played on a one-dimensional cellular field, as shown in Fig. 2 (field: 1–20). The players are homogeneous except in their starting positions (left team: player 1 (field 8), player 2 (field 5); right team: player 1 (field 13), player 2 (field 16)), and each player makes a run, dribbles the ball, makes a shot at goal, or passes the ball to a player of their own team. One of the actions is decided on based on the relative locations of all players and the ball (72 patterns). Action is taken in turn between the 2 teams. Each step in the game is composed of 4 actions by all players.

Multiple players can not be in one cell. The ball is always in a cell where a player resides. A moving action of a player with the ball means dribbling. Players move to either of the neighboring cells, but when a player moves to a cell with another player, the neighboring player is skipped over (a player cannot skip more than one other player). In this case, if the players are in opposite teams and one of them has the ball, the ball moves to the other player with a set probability (P_{steal}). If there is an opponent player between the passer and the receiver, the ball-passing becomes a failure with a set probability (P_{cut}), and in this case the ball moves to the cell where the oppenent player resides. The success rate for shooting is antiproportional to the length between the player's position and the goal irrespective of the presence of the opposing players. If a goal is scored, the game restarts with the initial player locations. If there is a failure, the game restarts after the ball is moved to the opposite player nearer to the goal post.

We expect two types of altruistic behavior which could lead to the emergence of cooperation in the game. One is

Table 1. Classification of the methods for evolving a team

Population structure			Each genome represents		Unit of fitness evaluation is		Code name
Depends on		Number of populations					
T?	R?						
No	No	1	A player	Heterogeneous players	A player	By direct evaluation	1–PHe–PD
						By indirect evaluation	1–PHe–PI
					A team (same fitness in a team)		1–PHe–T
				Homogeneous players	A team (same fitness in a team)		1–PHo–T
			A team	Fixed player-roles	A team		1–TFi–T
				Unfixed player-roles	A team		1–TUn–T
	Yes	R	A player	Heterogeneous players	A player	By direct evaluation	R–PHe–PD
						By indirect evaluation	R–PHe–PI
					A team (same fitness in a team)		R–PHe–T
Yes	No	T	A player	Heterogeneous players	A player	By direct evaluation	T–PHe–PD
						By indirect evaluation	T–PHe–PI
					A team (same fitness in a team)		T–PHe–T
				Homogeneous players	A team (same fitness in a team)		T–PHo–T
			A team	Fixed player-roles	A team		T–TFi–T
				Unfixed player-roles	A team		T–TUn–T
	Yes	$T \cdot R$	A player	Heterogeneous players	A player	By direct evaluation	TR–PHe–PD
						By indirect evaluation	TR–PHe–PI
					A team (same fitness in a team)		TR–PHe–T

T, number of teams in a game; R, number of player roles in a team

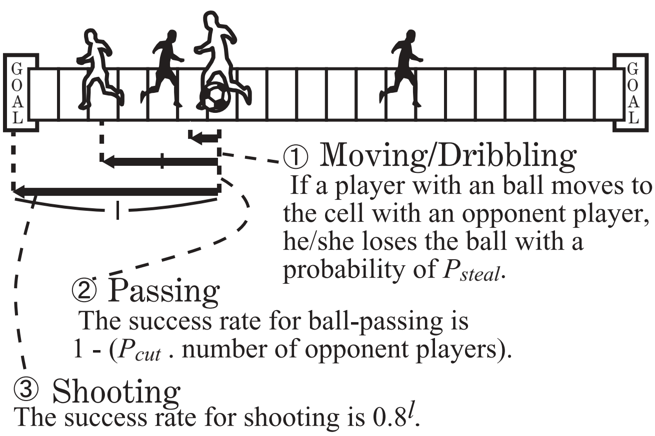

Fig. 2. The ultimately simplified soccer game

passing the ball to the other player in the same team instead of dribbling the ball or taking a shot at the goal. The other type is running in the direction away from the goal. The former type of altruistic behavior is analyzed in Sect. 4.3.

4 Evaluation

4.1 Expression of the players

Each player selects their next action deterministically based on the positional relationship of the players and the ball. In doing so, two opponents are not distinguished. So to be precise, the genetic information of each player decides the next action of that player based on one of 48 patterns, where each pattern is associated with one of the four actions: running/dribbling to the right; running/dribbling to the left; feeding (passing) the ball to a player of their own team; taking a shot at goal. Therefore, each player is represented by 96 bits of genetic information.

4.2 Evaluation setting

The evaluation is conducted in two steps: an evolution step and an evaluation step. In the evolution step, populations are evolved for 2000 generations using 18 methods independently. Each population has 40 individuals in all methods. A

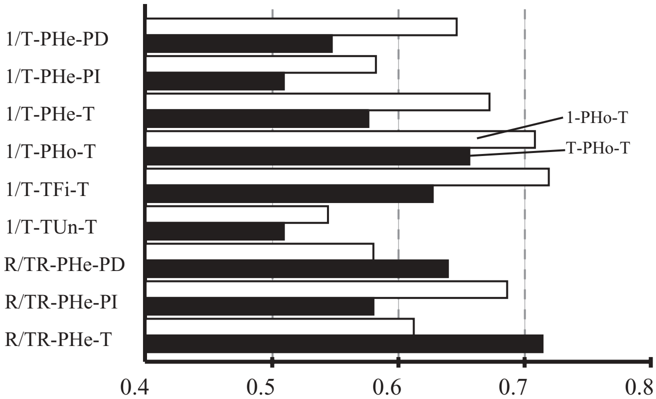

Fig. 3. The average winning ratio of the best 10 teams evolved by each of 18 methods

Table 2. Average winning ratio and assist ratio

Code name	Results			
	Winning ratio	Rank	Assist ratio	Rank
1–PHe–PD	0.673	7	0.146	15
1–PHe–PI	0.609	11	0.289	10
1–PHe–T	0.699	5	0.310	9
1–PHo–T	0.735	3	0.390	5
1–TFi–T	0.746	1	0.342	7
1–TUn–T	0.571	16	0.336	8
R–PHe–PD	0.607	12	0.109	16
R–PHe–PI	0.713	4	0.503	1
R–PHe–T	0.639	10	0.402	3
T–PHe–PD	0.574	15	0.080	17
T–PHe–PI	0.536	17	0.391	4
T–PHe–T	0.603	14	0.242	12
T–PHo–T	0.683	6	0.226	13
T–TFi–T	0.654	9	0.260	11
T–TUn–T	0.536	18	0.214	14
TR–PHe–PD	0.666	8	0.077	18
TR–PHe–PI	0.607	13	0.388	6
TR–PHe–T	0.741	2	0.416	2

round-robin tournament of an ultimately simplified game of 200 steps is held to evaluate the fitness in each generation.

The parameters P_{steal} and P_{cut} are set to 0.8 and 0.4, respectively, in both steps. These parameters were determined based on preliminary experiments mainly using [T–Pho–T], [T–Tfi–T], and [TR–Phe–T]. The evolution of the players depended significantly on both parameters. In short, a large P_{steal} or a small P_{cut} evolved the passer-type players. In contrast, a small P_{steal} or a large P_{cut} evolved the dribbler-type players. We found that the above settings could generate many different kinds of player.

With the <team-evaluated> option, the fitness is calculated as the number of goals the team scored minus the number of goals the opponent team scored. With the <direct-player-evaluated> option, the fitness is calculated as the number of goals the player scored minus the opposing team's goals divided by 2. Then tournament selection (repeatedly selecting the individuals with a higher fitness as parents by comparing two randomly chosen individuals), crossover with a 60% probability, and one-point mutation with a 3% probability are adopted as genetic operators. With the <indirect-player-evaluated> option, we use a primitive player designed a priori as follows. When a player keeps the ball, if they are behind the other team player they pass the ball to the other player, otherwise they shoot. When a player does not keep the ball, if they are behind the other team player, they move back, otherwise they move toward the goal. In the evaluation step, the best team is selected in each of the last 50 generations in the evolution step, and 50 × 18 selected teams conduct another round-robin tournament of 1000 step games.

4.3 Evaluation results

Figure 3 shows the winning ratio of the teams evolved by 18 methods, each of which is the average winning ratio of the best 10 teams from 50 teams in the all-play-all tournament described above. Table 2 (the left-hand column in the results) also shows these results. Each pair of bars shows the results of strategies with the same options in genome representation and fitness evaluation except for the population structure option (upper white bars, <1/R-population> options; lower black bars, <$T/T \cdot R$-populations> options).

It can be seen that the top three methods in this evaluation are <1-population, team-represented with fixed playerroles, team-evaluated>, <$T \cdot R$-population, heterogeneous-player-represented, team-evaluated>, and <1-population, homogeneous-player-represented, team-evaluated>. The winning ratios are 74.6%, 74.1%, and 73.5%, respectively. An additional evaluation using a team consisting of two primitive players showed that its winning ratio was 16.0%. This ratio could be a measure for the performance of these methods.

Regarding the population structure, <1/R-populations> options performed better than <$T/T \cdot R$-populations> options in general. This might be because of ill-balanced evolution, over-specialization, or circularity. The adoption of an asymmetric game as a testbed would make this tendency weaker. Regarding genome representation, the <homogeneous-player-represented> option performed well in general. Also, the <team-represented with fixed player-roles> option performed well, although the <team-represented with unfixed player-roles> option performed badly. Regarding fitness evaluation, the <team-evaluated> option performed well in general, as the fact that five of the top six methods adopt this option has shown. The performance of the <indirect-player-valuated> option depended largely on the other options.

We observed an interesting separation of roles between the two players in the teams with a high winning ratio. For example, in some teams the forward player tended to play near the goal and the backward player tended to move in order to intercept the ball, and in some teams both players seemed to use man-to-man defense.

Next, we examined the relationship between altruistic behavior which could lead to cooperative behavior and the winning ratio. Here we focus on the following behavior pattern. A player with the ball passes to the other team player, who receives the ball without being intercepted and then successfully shoots a goal immediately or after dribbling. We termed this series of actions an “assisted goal.” Table 2 shows the assist ratio, which is the ratio of assisted goals to all goals, and the winning ratio of the teams evolved by 18 methods. We can see from this table that good performing teams also have a tendency to have a high assist ratio. In contrast, it is not necessarily the case that teams with a high assist ratio have a tendency to have a high winning ratio. This means that the assisting behavior defined above is a necessary requirement for the teams to perform well.

It is a remarkable fact that the <indirect-player-evaluated> option made the assist ratio higher. In this option, we adopted a method in which the fitness of a player is the decrease in the fitness of team when that player is replaced by a primitive player. This method should generate a strong interaction between two players because it tends to result in a large decrease in team fitness when a player is replaced. Therefore, the teams generated by the indirect evaluation method have a higher assist ratio despite having a relatively low winning ratio.

5 Conclusion

This article has focused on methods for evolving a cooperative team by conducting a comprehensive evaluation of 18 methods. We have found that some methods perform well, while there are complex correlations among design decisions. Also, further analysis has shown that cooperative behavior can be evolved, and can be a necessary requirement for teams to perform well even in such a simple game. Future work includes a more detailed analysis of cooperative behavior, and an extension of the ultimately simplified soccer game.

References

1. Haynes T, Sen S, Schoenefeld D, et al. (1995) Evolving a team. Working Notes for the AAAI Symposium on Genetic Programming, MIT Press, Cambridge, MA, pp 23–30
2. Miconi T (2001) A collective genetic algorithm. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Morgan Kaufmann, San Francisco, CA, pp 876–883
3. Quinn M, Smith L, Mayley G, et al. (2002) Evolving formation movement for a homogeneous multi-robot system: teamwork and role allocation with real robots. Cognitive Science Research Paper 515, School of Cognitive and Computing Sciences, University of Sussex, Brighton
4. Luke S (1998) Genetic programming produced competitive soccer softbot teams for RoboCup97. Proceedings of the 3rd Annual Genetic Programming Conference (GP), Morgan Kaufmann, San Francisco, CA, pp 204–222
5. Potter MA , De Jong KA (1994) Cooperative coevolutionary approach to function optimization. Proceedings of the 3rd Parallel Problem Solving from Nature (PPSN), Springer Berlin, pp 249–257
6. Asai Y, Arita T (2003) Coevolution of morphology and behavior of robots in a multi-agent environment (in Japanese). Proceedings of the SICE 30th Intelligent System symposium, The Society of Instrument and Control Engineers, Tokyo, pp 61–66