A Comparative Evaluation of Methods for Evolving a Cooperative Team

Takaya Arita[1] and Yasuyuki Suzuki[2]
[1]Nagoya University, [2]Goldman Sachs Japan Holdings
Japan

1. Introduction

Some problems can be efficiently solved only by teams consisting of cooperative autonomous players (robots). Many researchers have developed methods that do not require human designers to define specific behaviors of players for each problem. The work reported in this chapter 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 (Haynes et al., 1995). 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.

This chapter is organized as follows. In Section 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 Section 3, we describe the ultimately simplified soccer game played on a one-dimensional field as a testbed for comparative evaluation of these 18 candidate methods. Section 4 reports on the results of the comparative evaluation of these methods, and Section 5 summarizes the work.

2. Methods for Evolving a Team

In general, 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 the 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 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 2 teams, with 3 populations corresponding 3 players, or with 6 populations corresponding to 2 teams of 3 players. So, the typical options for the number of

the populations are 1, *R*, *T* and *T*R* (*T*: number of teams in the game, *R*; number of the player roles in the team).
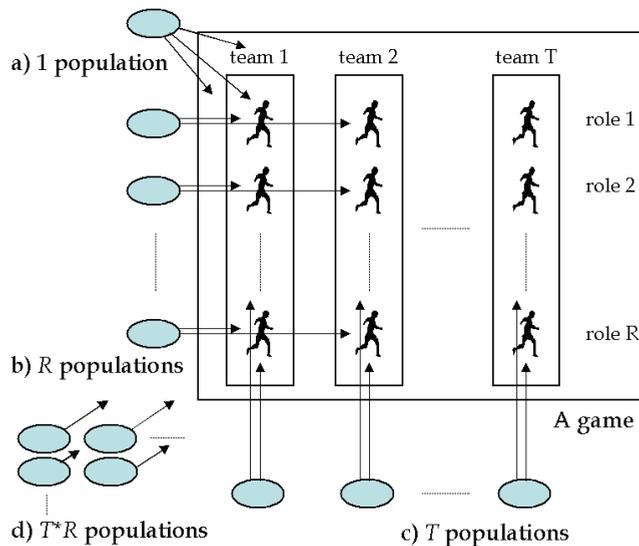

Figure 1. The four options for the population structure. a) The population represents all player roles in all teams. b) Each population represents one player role in all teams. c) Each population represents all player roles in each team. d) Each population represents one player role in each team

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 player 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 population. Some methods for indirect evaluation have been proposed (Miconi, 2001). We adopt a method as this option 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.

Population structure			Each genome represents		Unit of fitness evaluation is		Code name
depends on		Number of					
T?	R?	populations					
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*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

Table 1. Classification of the methods for evolving a team (T, number of teams in a game; R, number of player roles in a team)

Many researchers treated this issue, although most of them focused on one or two methods of 18 combinations. Some significant studies are classified into one of these combinations as follows: 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. (Quinn et al., 2002) adopted this method, and successfully evolved robots that work as a team, adopting and maintaining distinct but interdependent roles, based on their relative positions in order to achieve a formation movement task, although they were homogenous. Miconi (Miconi, 2001) 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 (Luke, 1998) 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 De Jong (Potter & De Jong, 1994) 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 (Asai & Arita, 2003) 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 comparative evaluation of these 18 candidate methods. It is a 2 vs 2 player game played on a one-dimensional cellular field ($T=R=2$ in Fig. 1), 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 the other 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 2 teams. Each step in the game is composed of 4 actions by all players.
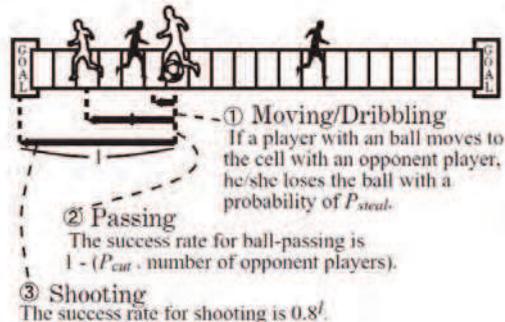

Figure 2. The ultimately simplified soccer game

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 opponent 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 moves 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 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 Section 4.3.

4. Evaluation

4.1 Expression of the Players

Each player selects next action deterministically based on the positional relationship of the players and the ball. In doing so, two opponent players 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 the other 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 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 players.

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 opponent 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 selected 50 times 18 selected teams conduct another round-robin tournament of 1000 step games.

4.3 Evaluation Results

Fig. 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 in Fig. 3 shows the results of the strategies with same options in genome representation and fitness evaluation except for the population structure option (upper white bars, <1/R-populations> options; lower black bars, <T /$T*R$-populations> options).

It can be seen that the top three methods in this evaluation are <1-population, team-represented with fixed player-roles, team-evaluated>, <$T*R$-populations, 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, the <1/R-populations> option performed better than the <T/$T*R$-populations> option 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 defence.
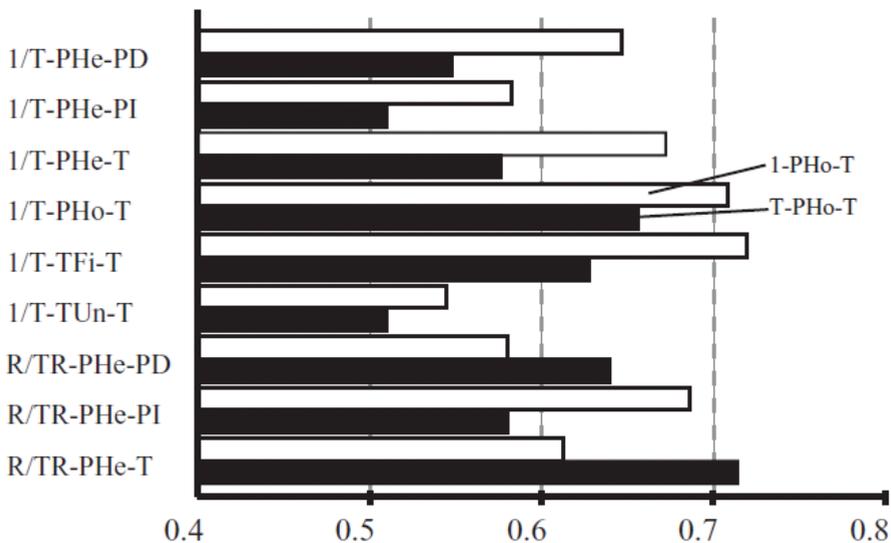

Figure 3. The average winning ratio of the best 10 teams evolved by each of 18 methods

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 as an "assisted goal" (Fig. 4). 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 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 assisting behaviour defined above is a necessary requirement for the teams to perform well.
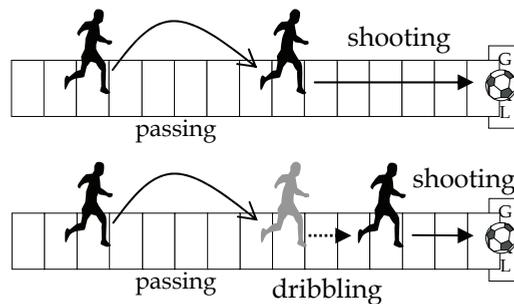
Figure 4. Two types of "assisted goals"

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

Table 2. Average winning ratio and assist ratio

It is a remarkable fact that the <indirect-player-evaluated> option made the assist ratio higher. For this option, we adopted a method in which the fitness of a player is the decrease in the fitness of team when the player is replaced by a primitive player. This method should generate a strong interaction between two players because it tends to result in large decrease when the 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 chapter has focused on the methods for evolving a cooperative team by conducting a comparative evaluation of 18 methods. We have found that some methods performed 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 the teams to perform well even in such a simple game. These results could provide insights into the evolutionary design of multi-robot systems working in cooperative tasks.

Evolutionary computation mimics the biological evolution of living organisms. We believe that the credit assignment problem, which is the focus of this chapter, could be solved more efficiently by developing the biological knowledge on the mechanism of the evolution of altruistic behavior, specifically on multi-level selection, a modern version of group selection (Ichinose & Arita, 2008).

6. References

Asai Y. & Arita T. (2003). Coevolution of Morphology and Behavior of Robots in a Multi-agent Environment, *Proceedings of the SICE 30th Intelligent System Symposium*, pp. 61-66, Tokyo, March 2003, The Society of Instrument and Control Engineers, Tokyo.

Haynes, T. ; Sen, S ; Schoenefeld, D. & Wainwright, R. (1995). Evolving a Team, *Working Notes for the AAAI Symposium on Genetic Programming*, pp. 23-30, Cambridge, MA November 1995, AAAI Press, Menlo Park, CA

Ichinose G. Arita, T. (2008). The Role of Migration and Founder Effect for the Evolution of Cooperation in a Multilevel Selection Context, *Ecological Modelling*, Vol. 210, Isuue 3, January 2008, pp. 221-230, ISSN 0304-3800

Luke, S. (1998). Genetic Programming Produced Competitive Soccer Softbot Teams for RoboCup97, *Proceedings of the Third Annual Genetic Programming Conference (GP)*, pp. 204-222, ISBN 1-55860-548-7, Madison, July 1998, Morgan Kaufmann Publishers, San Francisco

Miconi, T. (2001). A Collective Genetic Algorithm, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 876-883, ISBN 1558607749, San Francisco, CA, July 2001, Morgan Kaufmann Publishers, San Francisco

Potter, M. A. & De Jong, K. A. (1994). Cooperative Coevolutionary Approach to Function Optimization', *Proceedings of the Third Parallel Problem Solving from Nature (PPSN)*, pp. 249-257, Jerusalem, October 1994, Springer-Verlag, New. York

Quinn, M.; Smith, L.; Mayley G. & Husbands, P. (2002). Evolving Formation Movement for a Homogeneous Multi-Robot System: Teamwork and Role Allocation with Real Robots, *Cognitive Science Research Papers (CSRP)*, 515, School of Cognitive and Computing Sciences, University of Sussex, Brighton
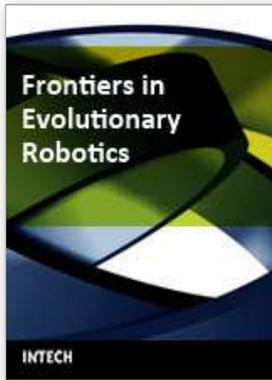
Frontiers in Evolutionary Robotics

Edited by Hitoshi Iba

This book presented techniques and experimental results which have been pursued for the purpose of evolutionary robotics. Evolutionary robotics is a new method for the automatic creation of autonomous robots. When executing tasks by autonomous robots, we can make the robot learn what to do so as to complete the task from interactions with its environment, but not manually pre-program for all situations. Many researchers have been studying the techniques for evolutionary robotics by using Evolutionary Computation (EC), such as Genetic Algorithms (GA) or Genetic Programming (GP). Their goal is to clarify the applicability of the evolutionary approach to the real-robot learning, especially, in view of the adaptive robot behavior as well as the robustness to noisy and dynamic environments. For this purpose, authors in this book explain a variety of real robots in different fields. For instance, in a multi-robot system, several robots simultaneously work to achieve a common goal via interaction; their behaviors can only emerge as a result of evolution and interaction. How to learn such behaviors is a central issue of Distributed Artificial Intelligence (DAI), which has recently attracted much attention. This book addresses the issue in the context of a multi-robot system, in which multiple robots are evolved using EC to solve a cooperative task. Since directly using EC to generate a program of complex behaviors is often very difficult, a number of extensions to basic EC are proposed in this book so as to solve these control problems of the robot.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Takaya Arita and Yasuyuki Suzuki (2008). A Comparative Evaluation of Methods for Evolving a Cooperative Team, Frontiers in Evolutionary Robotics, Hitoshi Iba (Ed.), ISBN: 978-3-902613-19-6, InTech, Available from: http://www.intechopen.com/books/frontiers_in_evolutionary_robotics/a_comparative_evaluation_of_methods_for_evolving_a_cooperative_team

INTECH

open science | open minds

www.intechopen.com