International Journal of
Science and Engineering Investigations
vol. 1, issue 9, October 2012
ISSN: 2251-8843

Evolving RoboCup Soccer Player Formations by means of Particle Swarm Optimization

Hidehiko Okada[1], Akira Yamashita[2], Tetsuya Wada[3]
[1,2,3] Graduate School of Frontier Informatics, Kyoto Sangyo University
([1] hidehiko@cc.kyoto-su.ac.jp)

***Abstract*-**Researchers have applied evolutionary algorithms to RoboCup Soccer players/teams and evaluate the effectiveness of the algorithm in evolving good players and teams, but further investigations are required to know more about the ability of the algorithms. We report our application of particle swarm optimization (PSO) to the evolution of RoboCup Soccer player formations: how well formations for various team performances (e.g., offensive, defensive, balanced) can be automatically obtained by means of PSO.

Keywords- *evolutionary algorithm; swarm intelligence; reinforcement learning.*

I. Introduction

Computational intelligence methods have been applied and evaluated to obtain capable autonomous agents. Researchers have applied evolutionary algorithms including genetic algorithm (GA) [1] to RoboCup Soccer [2] players/teams and evaluate the effectiveness of the algorithm in evolving good players and teams [3-7], but further investigations are required to know more about the ability of the algorithms. In this paper, we report on our application of particle swarm optimization (PSO) [8] to the evolution of RoboCup Soccer player formations: how well formations for various team performances (e.g., offensive, defensive, balanced) can be automatically obtained by means of PSO.

II. PSO Applied to Soccer Team Formations

There are five leagues in RoboCup Soccer, and our research is on the simulation league. A team consists of 11 software autonomous agents as players. As player codes, we utilize `AGENT2D` ver.2.1.0 [9], `HELIOS` for RoboCup JapanOpen2009 in Osaka [10] and `HELIOS` for RoboCup2010 in Singapore [11], all of which use the same format of player formation configuration files. These sample player codes utilize several configuration files for player formations (home positions) associated with the game states and the ball positions. Fig.1 shows the format of player position configuration in `normal-formation.conf`. Home positions of 11 players in the soccer field are specified with respect to each ball position. In the example of Fig.1, the home position of the 5th player is specified as (-11.56, 15.78) when the ball is at the center (0, 0) of the field. A phenotype and a genotype in our research are a team formation and a real vector, respectively. Values in `normal-formation.conf` are included in a team genotype vector. In total, a genotype in our research consists of 300 real values (= two dimensional (x, y) values ×10 players (the goalie is excluded from the position tuning) × 15 variations of ball positions), i.e., the search space is 300 dimensional. The 15 ball positions are shown in Fig.2 (five variations along with the horizontal x axis and three variations along with the vertical y axis).

```
Ball 0 0
1 -50 0
2 -15.53 -5.42
3 -15.53 5.42
4 -11.56 -15.78
5 -11.56 15.78
6 -6.73 -1.87
7 2.83 -10.81
8 2.83 10.81
9 9.3 -23.78
10 9.3 23.78
11 9.41 -3.12
```

Figure 1. Format of player home position configuration in `normal-formation.conf`.

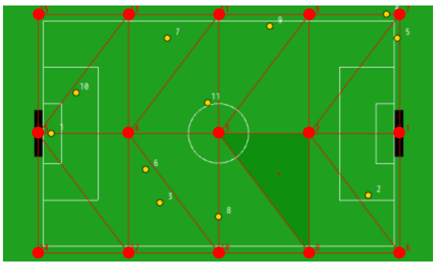

Figure 2. 15 ball positions (red points).

Fig.3 shows the flowchart of PSO in our research. First, each position (a set of 300 real values) X_i of a particle P_i is randomly initialized in the 300 dimensional search space, where i =1, 2, ..., N (N is the number of particles). Thus, X_i = { $x_{i,1}$, $x_{i,2}$, …, $x_{i,300}$ } and each $x_{i,j}$ is a real value. The velocity vector V_i = { $v_{i,1}$, $v_{i,2}$, …, $v_{i,300}$ } for each particle P_i is also randomly initialized, where each $v_{i,j}$ is also a real value. Then, a team T_i with the formation designated by X_i plays M games against benchmark teams B_j (j = 1, 2, ..., M). For example, the AGENT2D code is used for T_i and the HELIOS codes are used for B_j. Based on the goals scored in the benchmark games, fitness of each particle P_i (each team T_i with the formation designated by X_i) is evaluated. The personal best (*pbest*) of each particle and the global best (*gbest*) of the particles are updated based on the fitness scores, and then X_i and V_i of the particle P_i are updated:

$$V_i \leftarrow w \cdot V_i + c_1 \cdot r_1 \cdot (pX_i - X_i) + c_2 \cdot r_2 \cdot (gX - X_i),$$

$$X_i \leftarrow X_i + V_i,$$

where w is the inertia weight (0.9 in this research), c_1 and c_2 are constants (both 1.0 in this research), r_1 and r_2 are uniformly random values within the interval [0,1], pX_i is the *pbest* of P_i, and gX is the *gbest* of the particles { $P_1, P_2, \ldots, P_N$ }. The cycle in Fig.3 is iterated K times where K is a predefined number of cycles.

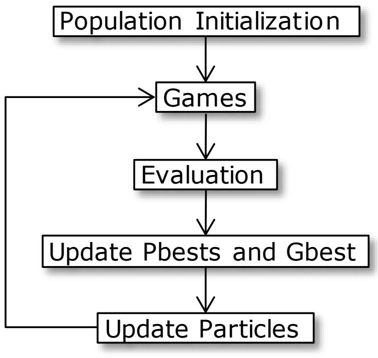

Figure 3. Flowchart of PSO applied to RoboCup Soccer in our research.

Teams (particles) are ranked as follows. A team which got more goals and lost fewer goals in the benchmark games is better. In our experiment, we tried single- and two-objective PSOs. In the experiment by the single objective PSO, the fitness score of a team T_i (let us denote the score as F(Ti)) is calculated as:

$$\mathrm{F}(T_i) = 0.5(\, \mathrm{G}'(T_i) + \mathrm{L}'(T_i)\,),$$

$$\mathrm{G}'(T_i) = \mathrm{G}(T_i) / \mathrm{G}(T_{i^*}),$$

$$i^* = \arg\max \mathrm{G}(T_i),$$

$$\mathrm{L}'(T_i) = 1 - (\, \mathrm{L}(T_i) / \mathrm{L}(T_{i^{**}})\,),$$

$$i^{**} = \arg\max \mathrm{L}(T_i),$$

where G(T_i) (or L(T_i)) is the mean goal score the team T_i got (or lost) in the benchmark games with the benchmark teams { $B_1, B_2, \ldots, B_M$ }. If G(T_{i^*}) = 0 then G'(T_i) is set to 0 for all i's. Similarly, if L($T_{i^{**}}$) = 0 then L'(T_i) is set to 1 for all i's.

In the experiment by the two-objective PSO, each team is ranked by the Pareto ranking method [12]. The two objectives are to get more/lose fewer goals. The scores of G(T_i) and L(T_i) are used as the fitness scores for the two goals respectively. The set of *gbests* (i.e., { gX_1, gX_2 … }) are non-dominated particles in { gX(1), gX(2), … }, where $gX(t)$ is the set of non-dominated particles at the t-th cycle. For each X_i, one of the *gbests* in { gX_1, gX_2 … } is selected to update V_i: a) if two or more *gbests* in { gX_1, gX_2 … } dominate X_i then one of such *gbests* is randomly selected, b) if only one *gbest* in { gX_1, gX_2 … } dominates X_i then the *gbest* is selected, and c) if no *gbest* in { gX_1, gX_2 … } dominates X_i then a *gbest* in { gX_1, gX_2 … } is randomly selected.

III. Simulation Results

A. Single-objective Optimization

We show three examples of simulation results by the single-objective PSO. Figs.4-6 show average scores of N teams over K iterations, where N = 200, 50, 50 and K = 30, 100, 40 respectively. In the simulation trials shown in Figs.4 and 5, the HELIOS 2009 code was used for { $T_1, T_2, \ldots, T_N$ } and the AGENT2D code was used for the benchmark team B_1 (i.e., M=1). In the simulation trial shown in Fig.6, the HELIOS 2010 code was used for { $T_1, T_2, \ldots, T_N$ } and the benchmark team was the same as for Figs.4 and 5. The values in normal-formation.conf were default ones for the AGENT2D benchmark team. A single game took about 2 mins, and thus the 6000 games (= 200 teams × 30 iterations) for the simulation in Fig.4 took about 12,000 mins (8.3 days) by using a PC with a 3.2GHz CPU (an AMD Phenom II X2 555 processor) and 2GB RAM.

In Figs.4-6, the { $T_1, T_2, \ldots, T_N$ } teams have become to get more goals (the offensive ability has been improved) over iterations. In Fig.5, the { $T_1, T_2, \ldots, T_{50}$ } teams have become to lose fewer goals (the defensive ability has been slightly improved) over iterations, but in Figs.4 and 6, the { $T_1, T_2, \ldots, T_N$ } teams do not have (the defensive ability does not have been improved). Similar results were obtained among parameter setting variations. Thus, for the three player codes used in our research, tuning the player formations is likely to contribute to improve the offensive ability better than the defensive ability.

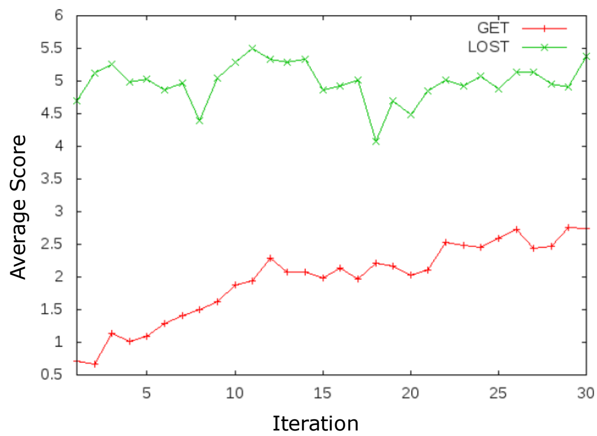

Figure 4. Average scores of 200 teams over 30 iterations.

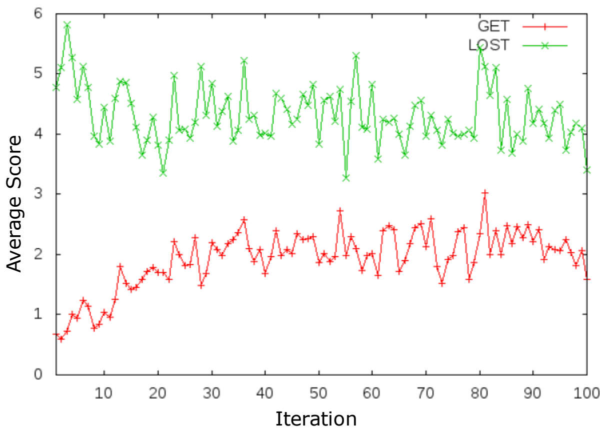

Figure 5. Average scores of 50 teams over 100 iterations.

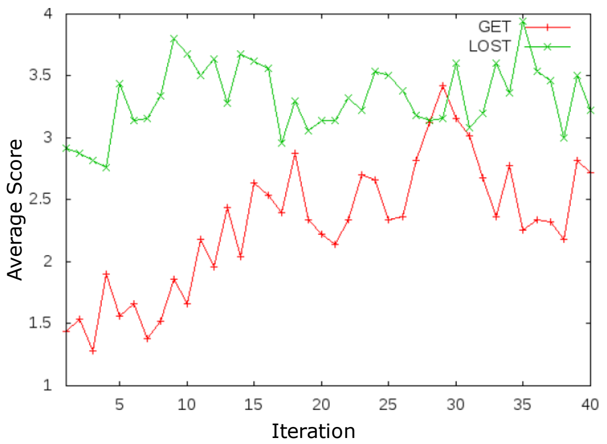

Figure 6. Average scores of 50 teams over 40 iterations.

Fig.7 shows an example of evolved player home positions of a team that got more goals and lost fewer goals in the simulation for Fig.4. In Fig.7, the red '+' points show player home positions with respect to the offensive ball positions (i.e., the ball is in the opponent team area), and the green '×' points show player home positions with respect to the defensive ball positions (i.e., the ball is in the team's own area). We expected that the home positions would likely be in the opponent team area for the offensive ball positions (to get goals) and the home positions would likely be in the team's own area for the defensive ball positions (to avoid losing goals). The player home positions in Fig.7 are not consistent with the expectation. The result in Fig.7 implies that midfield-based positioning is effective for both successful offense and defense.

B. Two-objective Optimization

We show an example of simulation result by the two-objective PSO. Fig.8 shows average scores of 50 teams over 83 iterations. In this simulation trial, the `HELIOS` 2009 code was used for { T_1, T_2, …, T_{50} } and two player codes were used for the benchmark teams B_1 and B_2 (i,e., M=2): B_1 was the `AGENT2D` code with a randomized `normal-formation.conf`, and B_2 was the `HELIOS` 2010 code with its default `normal-formation.conf`. The teams { T_1, T_2, …, T_{50} } were expected to be likely to get more goals against B_1 but lose more goals against B_2. In Fig.8, the 50 teams { T_1, T_2, …, T_{50} } have become to get more goals (the offensive ability has been improved) and lose fewer goals (the defensive ability has been slightly improved) over iterations. It was consistent among the results by the single/two-objective PSOs that the offensive ability could be improved more than the defensive ability could.

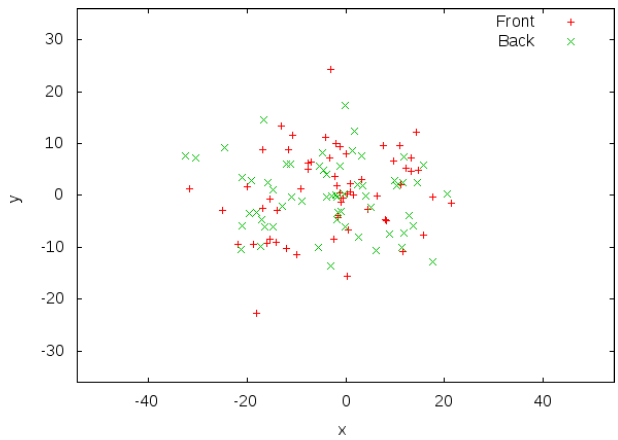

Figure 7. Player home positions of a team that got more goals and lost fewer goals.

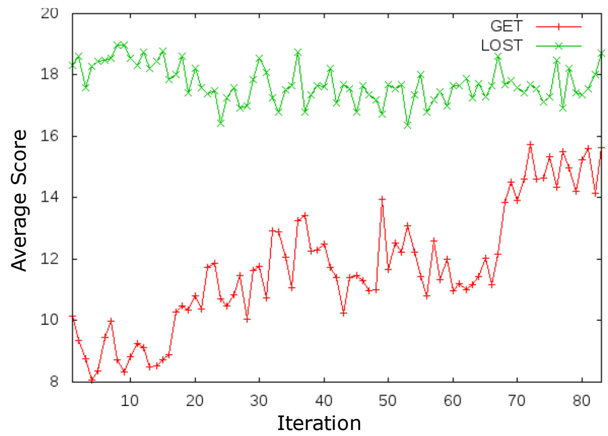

Figure 8. Average scores of 50 teams over 83 iterations.

Fig.9 shows performances of the 50 teams { T_1, T_2, …, T_{50} } in the first and the last (83th) generations. The red '+'/green '×' points illustrate the teams in the first/last generation. In Fig.9, the points of 50 teams move from the left area (in the first generation) to the right area (in the last generation), and thus the number of last teams that dominate first teams are more than the number of first teams that dominate last teams.

Fig.10 shows an example of evolved player home positions of a team that got more goals and lost fewer goals in the simulation for Figs.8 and 9. The red '+'/green '×' points illustrate the same information as in Fig.7. Again, the player home positions in Fig.10 are not consistent with our expectation described in the last section (III.*A*). The result in Fig.10 implies that well-balanced positioning is also effective for both successful offense and defense.

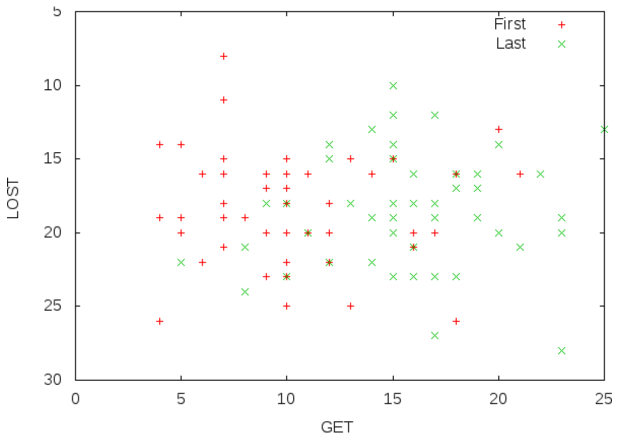

Figure 9. Performances of the 50 teams in the first/last generations.

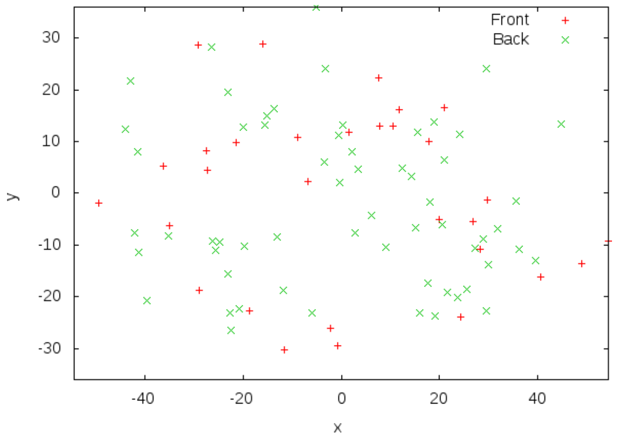

Figure 10. Player home positions of a team that got more goals and lost fewer goals.

IV. CONCLUSION

We applied single-/two- objective PSO to the evolution of RoboCup Soccer team formations for the simulation league. In this research, tuning the player home positions by means of PSO contributed to improve team offensive ability more than defensive ability. The evolved formations were not biased for offensive or defensive ball positions but were likely midfield-based or balanced in the game field.

In our future work, we should 1) further investigate whether these findings hold true for other player codes and other evolutionary methods, and 2) apply evolutionary methods not only to the player formations but to player roles and action rules.

REFERENCES

[1] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.

[2] http://www.robocup.org/robocup-soccer/ (last accessed on Oct. 12, 2012).

[3] S. Luke, C. Hohn, J. Farris, G. Jackson and J. A. Hendler, “Co-evolving soccer softbot team coordination with genetic programming,” Lecture Notes in Computer Science, Vol.1395 (RoboCup-97), 1998, pp.398–411.

[4] G. Dongbing, H. Huosheng, J. Reynolds and E. Tsang, “GA-based learning in behaviour based robotics”, Proc. of IEEE International Symposium on Computational Intelligence in Robotics and Automation, Vol.3, 2003, pp.1521–1526.

[5] T. Nakashima, M. Takatani, M. Udo and H. Ishibuchi, “An evolutionary approach for learning soccer strategies,” Proc. of Joint 2nd International Conference on Soft Computing and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems, 2004, 4pages, CD-ROM.

[6] V. Frias-Martinez and E. Sklar, “A team-based co-evolutionary approach to multi agent learning”, Proc. of the AAMAS Workshop on Learning and Evolution in Agent Based Systems, 2004, 6pages, CD-ROM.

[7] H. Okada and T. Takagi, “Evaluation of multi-objective genetic algorithm for RoboCupSoccer team evolution,” Proc. of International Conference on Instrumentation, Control and Information Technology (SICE Annual Conference), 2008, pp.151–154.

[8] J. Kennedy and R. Eberhart, “Particle swarm optimization,” Proc. of IEEE International Conference on Neural Networks, Vol.IV, 1995, pp.1942–1948.

[9] http://sourceforge.jp/projects/rctools/downloads/43082/agent2d-2.1.0.tar .gz/ (last accessed on Oct. 12, 2012)

[10] http://sourceforge.jp/projects/rctools/downloads/41357/helios-jopen2009 .tar.gz/ (last accessed on Oct. 12, 2012)

[11] http://sourceforge.jp/projects/rctools/downloads/48106/helios-10Singap ore.tar.gz/ (last accessed on Oct. 12, 2012)

[12] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, 2001.

Hidehiko Okada is currently a Professor with the Department of Computer Science and Engineering, Kyoto Sangyo University, Kyoto, Japan. He received the B.S. degree in industrial engineering and the Ph.D. degree in engineering from Osaka Prefecture University in 1992 and 2003, respectively. He had been a researcher with NEC Corporation from 1992 to 2003, and since 2004 he has been with the university. His current research interests include computational intelligence and human-computer interaction.

Akira Yamashita and **Tetsuya Wada** are currently graduate students at Graduate School of Frontier Informatics, Kyoto Sangyo University, Kyoto, Japan. They received the B.S. degrees in information communication engineering from Kyoto Sangyo University in 2011. Their research interests include evolutionary algorithms and their applications.