2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Cultural Algorithms Applied to the Evolution of Robotic Soccer Team Tactics: A Novel Perspective

Mostafa Z. Ali, Abdulmalik Morghem, Jafar Albadarneh, Rami Al-Gharaibeh, Ponnuthurai N. Suganthan, and Robert G. Reynolds

Abstract—**Cultural Algorithms have been previously employed to model the emergence of cooperative behaviors of agents in different multi-agent systems. In this paper, a simplified and adaptive version will be used as the basis to generate cooperative behaviors within a team of soccer players using different team formations and effective plays. This system can be used as a tutorial for the application of Cultural Algorithms for the coordination of groups of agents in complex multi-agent dynamic environments. Simplified Cultural Algorithms were successful in effectively learning different types of plays, including active and passive protagonists, within a small number of generations. Successful learning includes the coordination of adjustments of the team members to develop the most suitable team formations for every scenario. Experimental results enable us to conclude that Cultural Algorithms, when configured properly, in order to produce significant results, can perform very competitively when compared to other types of learning strategies and case-based game plays.**

I. Introduction

OVER the years, soccer simulation has turned into an attractive domain for applying different approaches of learning techniques in artificial intelligence (AI) and data mining [1]–[4]. This game can foster research in various fields using different technologies and techniques that can be examined and integrated to make an interesting and competitive play. When using such techniques, the goal is to develop optimal, or at least near-optimal, team and player behaviors and policies for this application.

Robocup [6]-[10] is the leading competition in this field as an international robotics competition to promote robotics and AI research. This international competition has many types of leagues; Humanoid, Standard size, Middle size, small size, and simulation league (2D, 3D). In this paper we are taking the 2D simulation track as a first step toward the 2D simulation league in Robocup.

Studying coordination and collaboration among the agents in a multi-agent system (MAS) is one of the most challenging, research directions in distributed AI. Different researchers have thoroughly studied different decision-making models [5] and new algorithms for new tactics (e.g. ball pass, handling, shoot and interception) [6]–[10] when competing in simulated soccer as an important application for evolving group behaviors between agents and coordinating their movement through the problem-solving process.

Previous works on Cultural Algorithms have examined the use of this type of evolutionary computation models to evolve cooperation within human cultures [11], [12]. Such work investigated the extent to which conceptions of cooperation and resource sharing can emerge between groups of individuals. In this paper, we investigate the use of a simplified and adaptive Cultural Algorithm to develop defensive and offensive plays and cooperative strategies among a team of autonomous robot soccer players. The evolution of such tactics are meant to adaptively enhance the team's play, ball control, and shot production instead of hard-coding and iteratively fine-tuning parameters based on the presented scenarios. An open source simulation system [1] is used to evaluate the proposed plans and effective skills for the team of robot players.

The rest of the paper is organized as follows: we begin by briefly describing the Cultural Algorithm approach in section II. Then in section III we discuss the current approaches from the literature that were employed to tackle similar type of landscapes. In section IV we give an overview of the simulation system in which we tested our team of robot players, a system that has the required simplicity for serving as a tutorial for concepts in agent technology and Cultural Algorithms. In section V we introduce the simplified adaptive version of Cultural Algorithms that is modified for this purpose. Section VI describes the experiments conducted within this framework. Section VII provides our conclusions.

II. Cultural Algorithms Configuration

A. Cultural Algorithms

Cultural Algorithm is a class of computational models derived from the cultural evolution process in nature [13], [14]. The pseudo-code for Cultural Algorithms is given in Fig. 1. $B(t)$ and $P(t)$ are the belief space and population space at time t. The algorithm starts by initializing the population and belief spaces and then enters the evolution loop, for a certain number of times, until the termination condition is satisfied.

At the end of a loop, each individual in $P(t)$ is evaluated using $F_{obj}()$. After individuals' fitness values are scored, an acceptance function $F_{acc}()$ is used to determine which of these individuals should update $B(t)$. The experiences of these accepted individuals will then be added to the belief space contents via $F_{update}()$. The newly generated and formed

Mostafa Z. Ali, Abdulmalik M. Morghem, Jafar AlBadarneh and Rami S. Al-Gharaibeh are with the Computer Information Systems department, Jordan University of Science and Technology, Irbid, 22110, Jordan, e-mail: mzali.pn@gmail.com, ammorghem09@cit.just.edu.jo, jwalbadarneh10@cit.just.edu.jo, rami@just.edu.jo.

Ponnuthurai N. Suganthan is an associate professor at the school of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, email: epnsugan@ntu.edu.sg

R. G. Reynolds is a professor at the Computer Science Department, Wayne State University, MI, 48202, USA, e-mail: reynolds@cs.wayne.edu

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

knowledge in $B(t)$ will influence the selection of the individuals for the next generation via $F_{\text{inf}}()$. A dual inheritance framework for the population and belief spaces is formed via the communication topologies, $F_{\text{acc}}()$,$F_{\text{inf}}()$, and $F_{\text{obj}}()$.

Begin
$t = 0$;
initialize $B(t)$, $P(t)$
while(!termination condition)
evaluate $P(t)$ { $F_{\text{obj}}()$}
update($B(t)$, $F_{\text{acc}}(P^t)$)
evolve($P(t)$, *influence*($B(t)$))
$t = t + 1$;
select $P(t)$ from $P(t\text{-}1)$
end
End

Fig. 1. Basic Pseudo-code for Cultural Algorithm

B. Knowledge Sources

Five basic knowledge sources have been identified in Cultural Algorithms. These knowledge sources include Normative, Situational, Topographic, Historic, and Domain knowledge sources.

Normative knowledge represents a set of promising parameter ranges. These variable ranges represent basics for individual behaviors and are used to guide individual adjustments [16]. This helps individuals progressing to good ranges of behavior.

Situational knowledge provides a set of representative cases of exemplar individuals. This knowledge is used for interpreting individuals' experiences. This type of knowledge helps individuals in imitating exemplars in the population.

Topographic knowledge makes use of region-based functional landscape patterns [17]. Based on spatial characteristics, topographic knowledge divides the whole functional landscape into cells. Each cell in the landscape keeps track of the best individual in this region, in a manner that emulates cell-best.

Historic knowledge observes all the events in the game and archives important events and scenarios. Among these events it records any change that occurs in the functional landscape that can be utilized as a case that can be used to reason about future moves.

Domain knowledge utilizes information from the problem domain to direct future search. For example, in a functional landscape represented in a soccer pitch, knowledge about the different types of regions in the pitch, goal region, where to go when the team possesses the ball and where to move when the team loses it should be useful in reasoning about them during the game in this dynamic environment.

These five knowledge sources were added at different times and are used to add different problem-solving capabilities to the Cultural framework. This set of knowledge sources is considered complete [16], [17] in the sense that any problem can be expressed as a subset of this set. When these knowledge sources are woven together, the interaction results in interesting collaboration behaviors in the soccer game.

III. Previous Work

Many previous works have assessed the use of evolutionary algorithms (EAs) to simulate the evolution of cooperation in human cultures [11], [12], [18]. The goal of this research has been to identify the limitations in information sharing and cooperation between agents in a multi-agent system.

Other researchers have investigated the use of EAs in dynamic environments, similar to those presented in the domain of RoboCup [19], [20]. This is a competition that is held every year and is considered as multidisciplinary research area that can be carried in a variety of implementations. These implementations support leagues composed of agents with varying capabilities [21], [22].

In [23] Mota et al. evolutionary algorithm used to manipulate location parameters and players' regions in the pitch. There, the authors used a genetic algorithm (GA) where the chromosome contained all the information for the agent's team. This technique makes tactical information available to all team members. The GA was always successful in finding a good solution. However, the algorithm needs many generations before its fitness ranking increases to a satisfactory level, and hence improving the team performance.

Hannebauer [8] proposed a methodology that makes use of a plan definition language to extract the representation of pertinent behaviors. The methodology helped to promote the reuse of such behaviors in future scenarios. The authors analyzed the behaviors which started from set-pieces and led to the scoring of goals while their team kept possession of the ball. The conclusions helped the authors to infer expressive rules to influence the process of generating rules from scratch.

De Raadt in [24] used a standard genetic algorithm to learn team strategies and set plays – set events that are extracted as a consequence for a specific situation. The evolutionary algorithm was used to optimize the behavior of every agent in the game. The extracted rules (using the proposed approach) were able to generate successful strategies as compared to the ones used by base team. Other techniques were used for such purpose to effectively learn set plays [15]. The authors in [25] used interaction nets to learn team strategies and find an optimal role assignment with a behavior that is similar to agent cooperation in RoboCup that arises between robots in the play field.

IV. Soccer Simulation Testbed

A. Prototyping Environment Motivation

In this section, we provide an overview of the simple soccer simulator and its main characteristics that are common to all prototypes and contestant teams across the various leagues. The simple soccer simulator was not the first of its type to be used for forming soccer teams and testing theoretical research in this field. First prototype approaches were attempted using the RoboCup simulator. The RoboCup soccer simulator is a rich environment for testing and developing soccer teams. On the other hand, its richness is what imposes extra constraints that made it less desirable for our research. As an example of the richness of the RoboCup simulator and its imposed constraints is that the simulated robot players are incapable of recognizing their teammates on the field. This feature is considered a constraint in that it forces the robot players to deal with perception. For the scope of this research, instead of spending much time on handling perception and object recognition the focus is on testing theories and tactics. Our utilized simulator could be used as a first stage for a RoboCup simulator implementation. Moreover, dealing with perception for theories at this stage will cause noise in the test data. This makes it harder to isolate the tested behaviors in a system that was meant as a brief tutorial for an application of CAs in dynamic environments.

B. *Overview of the simulator*

In a similar fashion to indoor soccer, the playing area in the simulator is a rectangular field enclosed by walls, as can be seen in Fig. 2. At each end of the field there is a goal for each of the two teams. The game consists of two opposing teams each with five players, one goalie and four field players. The game starts with a kick-off and continues until a goal is scored by either team, after which the ball is then replaced at the center of the field.. The game then resumes with another kick-off. A technical description of the implementation of the simulator is not relevant to this research and hence will not be provided here. More information can be found in [1].

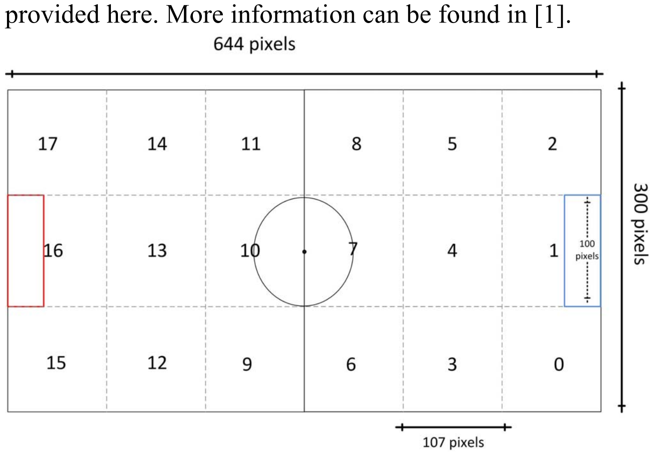

Fig. 2. Simple Soccer playing field

C. *Simple Soccer: Teams*

The team implemented in the simple soccer simulator is important to this research as it constitutes the base for the tested teams and implemented behaviors. The characteristics described below are basic ones and are possessed by all teams in the game. Any team consists of five players, one goalie and four field players, two of which are attackers and the other two are defenders. Steering behaviors and finite state machines used to implement basic heuristics in the game. Steering behaviors that are provided to players include:

- *Seek*: The player moves towards a target location without adjusting the speed of the player.
- *Separation*: A player is steered away from other players.
- *Arrive*: similar to seek but the player slows down as it approaches the target.
- *Pursuit*: the player treats the destination as a moving object (another moving player) and follows its direction.
- *Interpose*: finds out the midpoint between two objects and steers the player to that location.

A set of self-explanatory finite state machine (FSM) is utilized by field players as shown in Fig. 3. *GlobalPlayerState* is responsible for message routing. Messages are connected to the available states; as can be inferred from their names. These include *Msg_GoHome*, *Msg_ReceiveBall*, *Msg_PassToMe*, *Msg_Wait*, *Msg_SupportAttacker*. The message determines the player's next state during the game play.

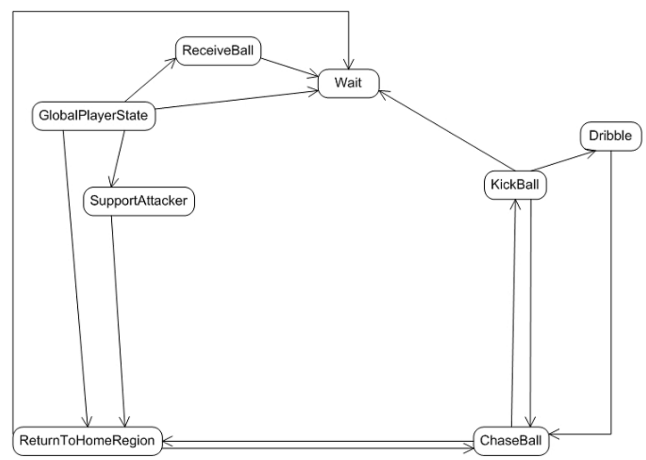

Fig. 3. FSM for any team in the simple soccer simulator [1]

The *ReceiveBall* state directs players to either arrive at the ball's target or to pursuit the ball. The decision is determined based on some factors, or might happen randomly. Another factor that affects this state is the presence of an opponent player within a threatening radius. *KickBall* is a state that will be entered when the player attempts to shoot at the goal or pass to another player. *ReturnToHomeRegion* causes all players to return to their predefined home regions. For each team there is a goalie, left attacker and right attacker, left defender and right defender. More information is provided in [1].

V. Evolving Tactics Using Cultural Algorithms

A. *Enhancing the offense*

We will now discuss the framework that is used to train one of the teams in the game based on the overview on CA as presented in previous sections. The fitness is the number of goals scored by the team. Any population-based evolutionary algorithm can be used for the population space. Examples are GA, evolutionary programming (EP) and particle swarm optimization (PSO) [16], [17]. We selected the classic EP configuration of the CA for the population model. The pseudo-code of the overall algorithm is given in Fig. 4. For each individual in the population, there are actions that can be taken and regions that it can occupy. Each player will have a state (one from those previewed in Fig. 3), and a region. For the entire team, an individual might be represented as:

[P1R|P1S|P2R|P2S|P3R|P3S|P4R|P4S]

As can be seen in Fig. 4, the default regions and states are predefined and assigned to all players in both teams. For the red team, players will be assigned to regions 16, 3, 5, 9, 11, and for the blue team players are situated in regions 1, 12, 14, 6, 4. States for all players will be initialized to prepare for kick-off. The goal is to train the team to choose the best regions and best states depending on the scenario whether offending or defending.

The belief space is simplified and has only generalizations of the behaviors (regions and states) associated with plans that exhibit the best performance. Such beliefs normally circumscribe the behaviors that an individual can select from, and are represented in terms of intervals. The mutation step is based on the resultant limits. These values for intervals are the base for affecting the mutation operator.

The simulation run continues for 150 generations where the algorithm checks the status of some indicators every 15 generations (epoch). If the average score of an epoch (*avg*) superseded the previous epoch (*pre*) and the ratio of scores for red to blue is less than one, then we will record the best regions when the goal was scored by the red team. Moreover, the states that players entered while moving around till reaching the goal regions clearly affect the status of the team. These recorded regions and states are used for future generations in developing a good plan for starting with when playing online (without having the learning feature turned on). The best behaviors are extracted from the produced log files from the game after 150 generations as described above.

Fig. 5 shows how the soccer robot players change their positions effectively while approaching the goal of their opponent. This should also enhance their supporting positions to the player possessing the ball.

B. Enhancing the defense

Defensive plans help the team reduce the number of scored goals. Defense is more important than offense. Reducing the number of scored goals by the opponent will enhance the rank of the team even with basic offensive plans. One of the best techniques a team should be trained on is to identify the best time to place a player between an opponent possessing the ball and his supporting player. This way, the player is enforced to either shoot the ball or continue on his own which decreases the number of scored goals.

Begin
  $t = 0$;
  initialize Avg, pre, ratio, best
  initialize plan {**Red**: 16,3,5,9,11; **Blue**: 1,12,14,6,4}
  initialize $B(t)$
  initialize $P(t)$ of N candidate solutions
  evaluate $P(t)$
  while($Gen \leq 150$)
    Copy parents to create N children
    Mutate each child
    evaluate $P(t)$ { $F_{\text{obj}}()$}
    select the best N individuals
    update($B(t)$, $F_{\text{acc}}(P^t)$)
    discard the poorest N individuals
    if(Gen % 15 == 0)

$$avg = \sum_{i=1}^{15} score_i$$

$$ratio = \sum_{i=1}^{15} score_i(red) \Big/ \sum_{i=1}^{15} score_i(blue)$$

      if(Avg>pre) and ($ratio$<1.0)
        Record *best* (*score*)
        Plan = {**Red**: *best*}
        adjust($B(t)$, accept(*best*))
        update($B(t)$, F_{acc}(*best*))
      end
    end
    $t = t + 1$
  end
End

Fig. 4. Pseudo-code for the adaptive Simplified Cultural Algorithm framework

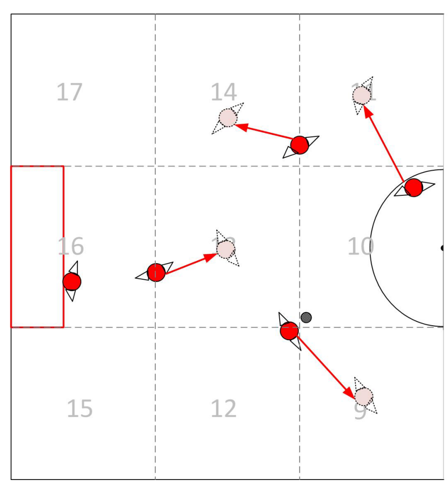

Fig. 5. Enhancing positions of players by chooses more appropriate regions to support the attacker and approach the goal

This simple, yet very effective, plan is illustrated in Fig. 6. The crucial aspect is to train the players to anticipate when to interpose themselves between the opponent who has the ball and his supporter waiting to receive the pass. When the opponent with the ball reaches beyond a certain point, it is better to start interposing in order to reduce the load on the game simulator, and switching states.

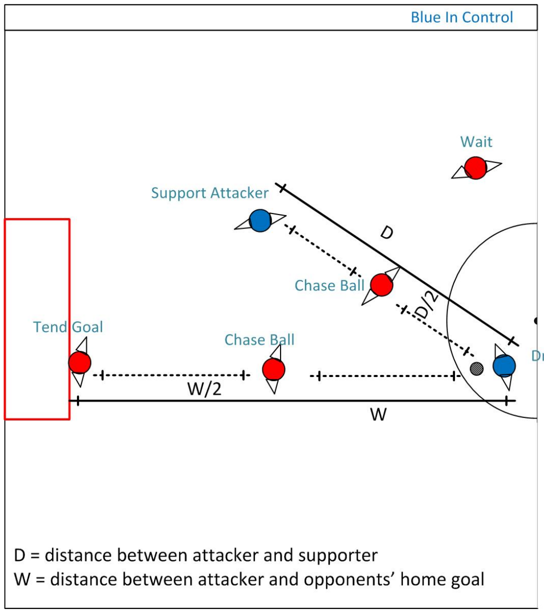

Fig. 6. Enhancing defense: interposing the opposing player with the ball and his supporting player

Defense is one of the two team states. When the team enters this state, every player should learn in which region to place itself and how to interpose the opponent players.

VI. Experimental Results and Analysis

In all of the experiments described in this section, the red team represented our modified team. Our results come from plays made against several other representative teams, including teams that play offensively and teams that play defensively. Moreover, our team played against the default team, and the GA team. In all of the runs, we will show the results from training the team on the presumed skills and plans. Each training session is 150 generations for which we conducted 10 runs. Training sessions were usually composed of 2 stages. The results of the first stage (with a focus on enhancing the offense) are used as a starting point for the second stage. With a two-stage learning we initialize all simulation variables and time in the second stage and see how the learning curve goes with respect to the first stage and using its resultant tactic as a starting point. Finally, we will present the results of playing our team against all other teams in 35 runs, where each run is 10 minutes long. The conducted simulations with their results are discussed next.

Experiment 1. *Proposed CA-team against the default team*

In this experiment, the CA-team played against the default team with basic skills as previously described. It was evident that the enhancements added using a strong defense, EP-based offense achieved superior results compared to those obtained using the default team. Fig. 7 shows the results out of 10 runs, for each of 150 generations in the first stage, training the team to be able to adaptively pick the best region while playing with a proper state. The figure shows the best, mean, and worst curves among the 10 runs for the proposed CA team and the default team. The difference in terms of score is evident. The second stage of training, Fig. 8 shows that although scoring of red team is lower, yet it was successful in selecting the most appropriate time to cut the opponent's passes.

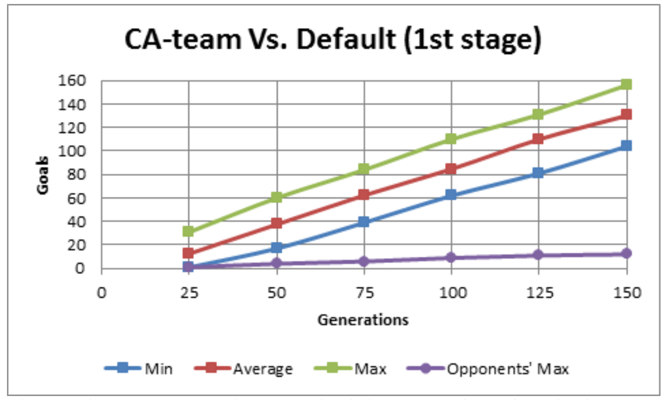

Fig. 7. The average, maximum and minimum number of goals that were recorded in 10 runs, each of 150 generations long for a game between CA-team and the default team (1st learning stage)

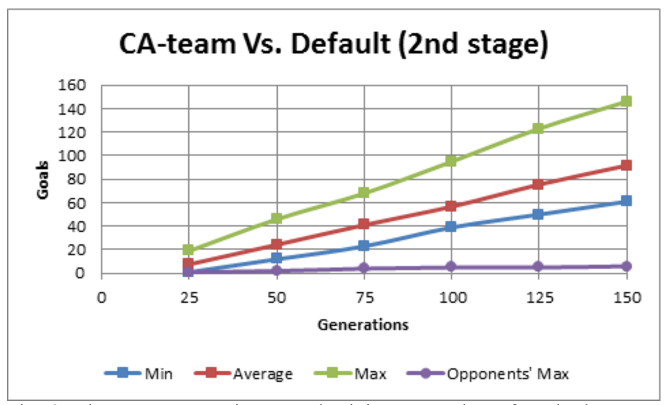

Fig. 8. The average, maximum and minimum number of goals that were recorded in 10 runs, each of 150 generations long for a game between CA-team and the default team (2nd learning stage)

Experiment 2. *Proposed CA-team against the team with a strong offense*

In this experiment, the red team played against a team that is trained to enhance only its offense using regular EP. As results show in Fig. 9, first stage training represented a very strong offense using CA compared to basic EP. In Fig. 10, the performance of the red team is enhanced with a stronger defense (needs more learning time than offense). It is worth noting that the performance of the EP-based team that enhances the offense using only best regions is better than that of the default team. This is clear from the resultant score of games between our algorithm from one side and the EP-based team from the other side.

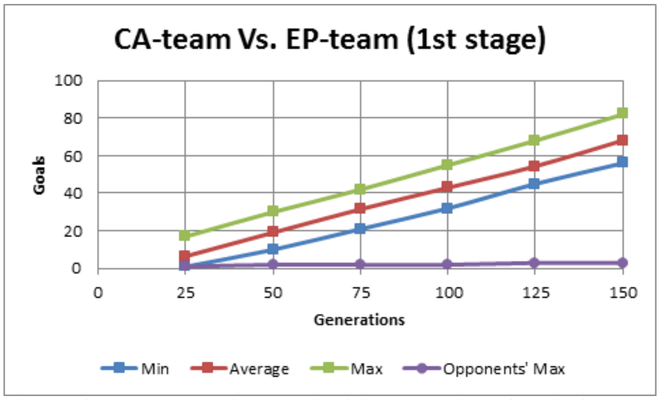

Fig. 9. The average, maximum and minimum number of goals that were recorded in 10 runs, each of 150 generations long for a game between CA-team and the EP team (1st learning stage)

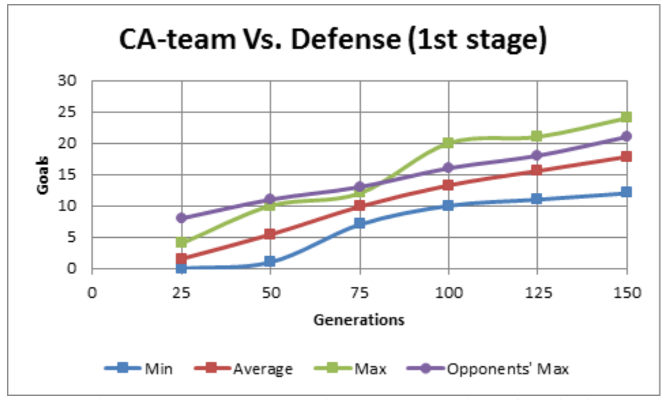

Fig. 11. The average, maximum and minimum number of goals that were recorded in 10 runs, each of 150 generations long for a game between CA-team and the strong defense team (1st learning stage)

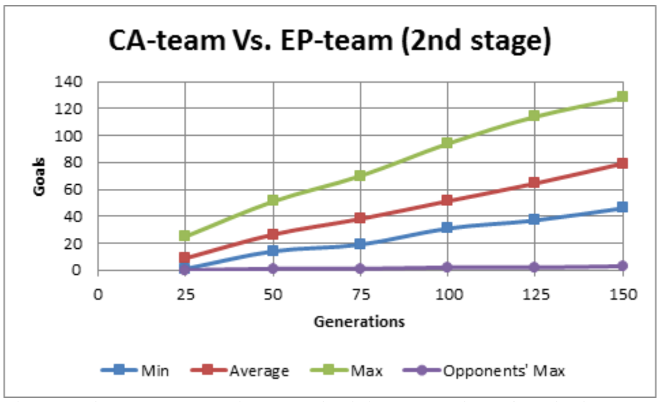

Fig. 10. The average, maximum and minimum number of goals that were recorded in 10 runs, each of 150 generations long for a game between CA-team and the EP team (2nd learning stage)

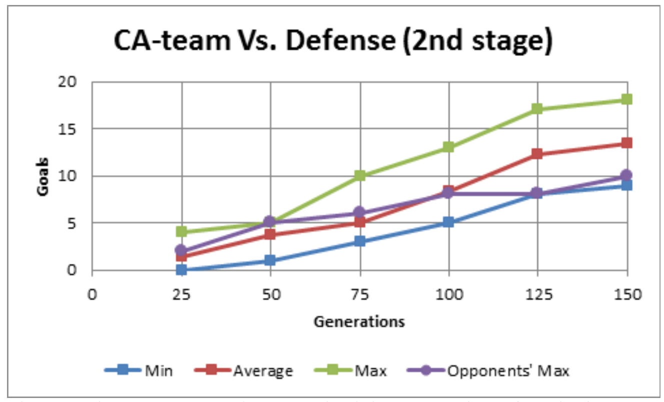

Fig. 12. The average, maximum and minimum number of goals that were recorded in 10 runs, each of 150 generations long for a game between CA-team and the strong defense team (2nd learning stage)

Experiment 3. *Proposed CA-team against the team with a strong defense*

The red team was made to play against another team with a stronger defense. This opposing team always places a player between our player who possesses the ball and its supporting player as soon as our team takes control. This is supposed to be more aggressive in stopping any pass that can make a potential plan for scoring a goal. Hence, this made it harder to select the proper plan and timing to avoid such tactic. As soon as the red team was trained to play against such stronger defense team it was able to find better plans and states. The collective behaviors of all players in terms of proper states and better regions were able to evolve successful tactics that led to winning. Fig. 11 shows how our algorithm was able to produce better scores. Stronger defense team (opposing team) was close and came between the best and average among our learning curves during the first learning stage.

The overall performance of the algorithm became better during the second learning stage. There is an apparent difference between the best scenarios of both techniques as seen in Fig. 12.

Experiment 4. Proposed CA-team against the GA-team

In this experiment the CA-team played against a GA model that used tournament selection. The belief space in CAs helps utilizing more function evaluations in searching for better plans. Fig. 13 shows that the results obtained from stage 1 training make it clear that the performance of CA-team is better than that of the GA team. Although the score of the CA-team in the second stage is not higher than that of the first stage but the GA team was not able to score as much goals as in the first stage. This is shown in Fig. 14.

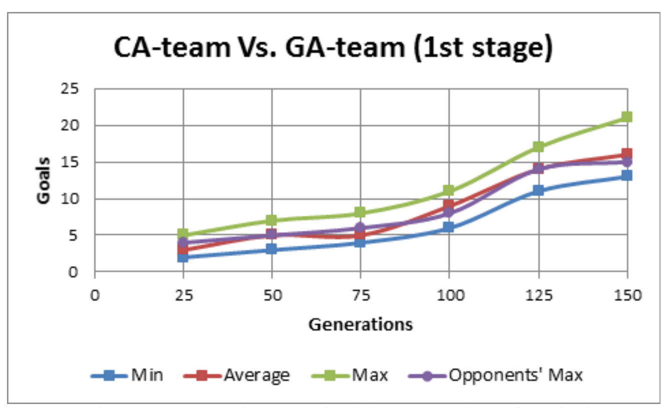

Fig. 13. The average, maximum and minimum number of goals that were recorded in 10 runs, each of 150 generations long for a game between CA-team and the GA- team (1st learning stage)

The results of the 35 runs for experiments 1–4 are shown in Fig. 15. In this figure, a peak means that there is positive difference in the score between the two teams to our favor. While the trough of a wave indicates that there is a negative difference and hence it means that we lost that game. Small amplitudes mean that the difference, if it exists, is small.

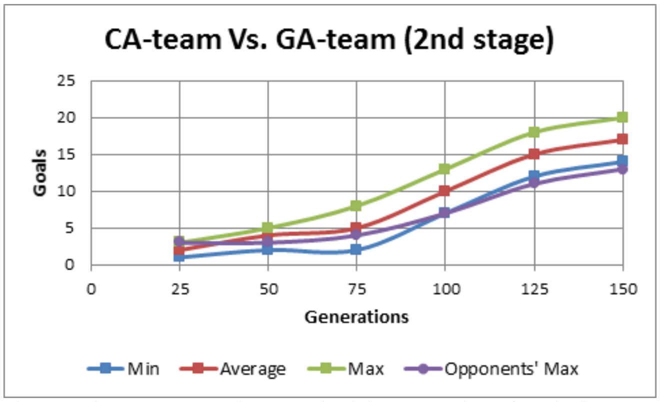

Fig. 14. The average, maximum and minimum number of goals that were recorded in 10 runs, each of 150 generations long for a game between CA-team and the GA- team (2^{nd} learning stage)

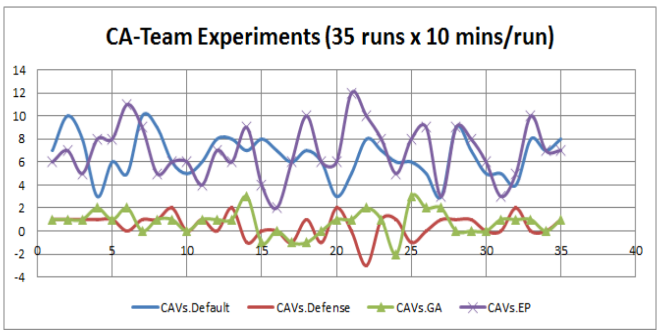

Fig. 15. Difference in scoring between the different pairs of experiments

A summary for some important statistics for all the experiments is given in Table 1. Table 2 summarizes all statistics for all types of experiments at generation 25, 50, 75, 100, 125 and 150.

TABLE 1
PERFORMANCE OF THE ALGORITHM AGAINST ALL OTHER TEAMS

Game description (35X10mins)	Wins	Losses	Draw	Avg. goals	Avg. opponent's goals
(Experiment 1) CA Vs. Default	35	0	0	6.571428	0.057142857
(Experiment 2) CA Vs. EP	35	0	0	7.314285	0.428571429
(Experiment 3 CA Vs. Defense	19	5	11	0.885714	0.428571429
(Experiment 4) CA Vs. GA	23	4	8	1.002157	0.389236801

VII. CONCLUSION

The Adaptive Cultural Algorithm framework with an embedded EP component as a population space was able to successfully learn different types of plays. The system utilized the belief space to record important events to be able to fine tune the ranges of individuals' behaviors. The system was able to learn the best regions and states to be assumed by the team at offensive and defensive situations. These situations were demonstrated through the use of opponents with different skill sets. Other useful environmental parameters and attributes related to the play domain could be useful in finding stronger tactics for obtaining higher scores. This will be investigated in future work.

REFERENCES

[1] Buckland, M., Programming Game Ai by Example. Plano, Texas: Worldware Publishing Inc., 2005.

[2] Wang, C.; Chen, X.; Zhao, X.; and Ju, S. "Design and Implementation of a General Decision-Making Model in Robocup Simulation," International Journal of Advanced Robotic Systems, vol. 1, no. 3, pp. 207-212, 2004.

[3] Dashti, H.; Aghaeepour, N.; Asadi, S.; Bastani, M.; Delafkar, Z.; Disfani, F.; Ghaderi, S.; and Kamali, S. "Dynamic Positioning Based on Voronoi Cells (Dpvc)," in Robocup 2005: Robot Soccer World Cup Ix ed. , vol. 4020, A. Bredenfeld, Jacoff, A., Noda, I., Takahashi, Y., Eds. Heidelberg, Germany: SBH, 2006, pp. 219–229.

[4] Almeida, F.; Abreu, P.H.; Lau, N.; and Reis, L.P. "An Automatic Approach to Extract Goal Plans from Soccer Simulated Matches," Soft Computing, vol. 17, no. 5, pp. 835–848, 2013.

[5] Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; Abd, H.; and Matsubara, E.O. "Robocup: A Challenge Problem for AI," AI Magazine, vol. 18, no. 1, pp. 73–85, 1997.

[6] Kyrylov, V.; Greber, M.; Bergman, D. "Multi-Criteria Optimization of Ball Passing in Simulated Soccer," Journal of Multi-Criteria Decision Analysis, vol. 13, pp. 103–113, 2005.

[7] Mota, L.; Reis, L.P.; and Lau, N. "Multi-Robot Coordination Using Setplays in the Middle-Size and Simulation Leagues," Mechatronics, vol. 21, no. 2, pp. 434–444, 2011.

[8] Hannebauer, M.; Wendler, J.; and Pagello, E. Eds., Balancing Reactivity and Social Deliberation in Mas – from Robocup to Real-World Applications, Vol. 2103, Heidelberg, Germany: SBH, 2001.

[9] Abreu, P. H.; Silva, D. C.; Mendes-Moreira, J.; Reis, L. P.; and Garganta, J. "Using Multivariate Adaptive Regression Splines in the Construction of Simulated Soccer Team's Behavior Models," International Journal of Computational Intelligence Systems, vol. 6, no. 5, pp. 893–910, 2013.

[10] Abreu, P. H.; Moura, J.; Silva, D. C.; Reis, L. P.; and Garganta, J. "Performance Analysis in Soccer: A Cartesian Coordinates Based Approach Using Robocup Data." Soft Computing, vol. 16, no. 1, pp. 47–61, 2012.

[11] Reynolds, R. G. "On Modeling the Evolution of Hunter-Gatherer Decision-Making Systems," Geographical Analysis, vol. 10, no. 1, pp. 31-46, 1978.

[12] Reynolds, R. G. "An Adaptive Computer Model of the Evolution of Agriculture for Hunter-Gatherers in the Valley of Oaxaca, Mexico.," Ph.D. dissertation, Dept. of CS, Univ. of Michigan, Ann Arbor, MI, 1979.

[13] Reynolds, R. G. "An Introduction to Cultural Algorithms," In Proc. Third Ann. Conf. on Evolutionary Programming, San Diego, CA, 1994, pp. 131-139.

[14] Reynolds, R. G. "Learning to Cooperate Using Cultural Algorithms." In Simulating Societies, Nigel Gilbert and J. Doran Eds. London, UK: UCL Press, 1994, pp. 223–244.

[15] Stone, P.; Sutton, R.; and Kuhlmann, G. "Reinforcement Learning for Robocup-Soccer Keepaway," Adaptive Behavior, vol. 13, no. 3, pp.165–188, Sep 2005.

TABLE II
STATISTICAL RESULTS FOR THE EXPERIMENTS

Training Set	Generations	1st Stage				2nd Stage			
		Min	Avg	Max	Opp Max	Min	Avg	Max	Opp Max
CA-team Vs. Default	25	1	12.56	31	1	1	7.64	19	1
	50	17	37.46667	60	4	12	24.32	46	2
	75	39	62.30667	84	6	23	41.13333	68	4
	100	62	84.73333	110	9	39	56.70667	95	5
	125	81	109.96	131	11	50	75.50667	123	5
	150	104	130.12	156	12	61	91.66667	146	6
[(Default)Opp. Max / CA-team] * 100		7.69 %				4.11 %			
CA-team Vs. EP-team	25	1	6.493333	17	1	1	8.88	25	0
	50	10	19.09333	30	2	14	26.46667	51	1
	75	21	31.54667	42	2	19	38.08	70	1
	100	32	43	55	2	31	51.30667	94	2
	125	45	54.37333	68	3	37	64.45333	114	2
	150	56	68.09333	82	3	46	79.2	128	3
[(EP-team)Opp. Max / CA-team] * 100		3.66 %				2.34 %			
CA-team Vs. D-team	25	0	1.546667	4	8	0	1.44	4	2
	50	1	5.426667	10	11	1	3.733333	5	5
	75	7	9.866667	12	13	3	5.053333	10	6
	100	10	13.28	20	16	5	8.306667	13	8
	125	11	15.57333	21	18	8	12.29333	17	8
	150	12	17.82667	24	21	9	13.38667	18	10
[(D-team)Opp. Max / CA-team] * 100		87.5 %				55.56 %			
CA-team Vs. GA-team	25	2	3	5	4	1	2	3	3
	50	3	5	7	5	2	4	5	3
	75	4	5	8	6	2	5	8	4
	100	6	9	11	8	7	10	13	7
	125	11	14	17	14	12	15	18	11
	150	13	16	21	15	14	17	20	13
[(GA-team)Opp. Max / CA-team] * 100		71.42857 %				65 %			

[16] Ali, M. Z.; and Reynolds, R. G. "An Intelligent Social Fabric Influence Component in Cultural Algorithms for Knowledge Learning in Dynamic Environments, Web Intelligence and Intelligent Agent Technology," In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '09), vol. 2, Milan, Italy, 15-18 Sept 2009, pp. 161-168.

[17] Chung, C.; and Reynolds, R. G. "CAEP: An Evolution-Based Tool for Real-Valued Function Optimization Using Cultural Algorithms," International Journal on Artificial Intelligence Tools, vol. 7, no. 3, pp.239-291, 1998.

[18] Faria, B. M.; Reis, L. P.; Lau, N.; and Castillo, G. "Machine Learning Algorithms Applied to the Classification of Robotic Soccer Formations and Opponent Teams." In IEEE International Conference on Cybernetics and Intelligent Systems (IEEE CIS), Singapore, 28-30 June 2010, pp. 344-349.

[19] Jin, X.; and Reynolds, R. G. "Using Knowledge-Based Evolutionary Computation to Solve Nonlinear Constraint Optimization Problems: A Cultural Algorithm Approach," In Congress on Evolutionary Computation (CEC 99), vol. 3, Washington DC, Jul 1999, pp. 1672-1678.

[20] Kvasnicka, V.; Pospíchal, J.; and Tino, P., Evolucné Algoritmy. Slovakia, Bratislava: STU, 2000.

[21] Lau, N.; Lopes, L. S.; and Corrente, G. A. "Cambada: Information Sharing and Team Coordination," In Proc. Eighth Conference on Autonomous Robot Systems and Competitions, Aveiro, Portugal: Universidade de Aveiro, April 2008, pp. 27-32.

[22] Lekavy, M. "Optimising Multi-Agent Cooperation Using Evolutionary Algorithm," In Proc. Student Research Conference in Informatics and Information Technologies (IIT. SRC 2005), M. Bielikova, Eds. Slovakia, Bratislava: Faculty of IIT, STU, April 2005, pp. 49-56.

[23] Mota, L.; Lau, N.; and Reis, L.P. "Co-Ordination in Robocup's 2d Simulation League: Setplays as Flexible, Multi-Robot Plans," In IEEE Conference on Robotics Automation and Mechatronics (IEEE RAM), Singapore, 28-30 June 2010, pp. 362-367.

[24] De Raadt, M.; Prokopenko, M.; and Butler, M. "Evolving tactical formations on the RoboCup field," In Electronic Proceedings of the Workshop on Adaptability in Multi-Agent Systems at The First RoboCup Australian Open, January 2003.

[25] Zweigle, O.; Lafrenz, R.; Buchheim, T.; Kappeler, U.-P.; Rajaie, H.; Schreiber, F.; and Levi, P. "Cooperative Agent Behavior Based on Special Interaction Nets," Intelligent Autonomous Systems: IAS-9, pp. 651-659, 2006.