ICS'02 Workshop On Artificial Intelligence

The Concepts Of The SoccerTeam Prototype

Bora I. Kumova*; Akira Imada+

*The University of Birmingham; School of Computer Science; Birmingham; UK
B.I.Kumova@cs.bham.ac.uk; http://cs.bham.ac.uk/~bik
Fax.: +44−121−414 2794; Fax.: +44−121−414 4281
+Brest State Technical University; Brest; Belarusia

Abstract. Planning in multi−agent systems is still a challenge task, especially for heterogeneous communication partners. While we present the major concepts of the SoccerTeam prototype, we discuss the implementation of the co−evolutionary planning (CEVOP) methodology for building individual and team plans consisting of primitive and combined Soccer game strategies.

Keywords. Planning; Evolutionary Computation; Distributed AI; Intelligent Agents; Knowledge−based Systems

1 Introduction

Evolutionary techniques in multi−agent systems [1] are mainly utilised for reducing exhaustive search in related problem domains, such as for planning or learning through exploring the related search space. For instance for co−operative learning [4]. Co−evolutionary algorithm enable further to evolve co−operative and/or competitive structures for multi−agent systems [3] or to evolve a whole multi−agent system according to some domain rules [11], [12]. However, all approaches consider either co−operation or competition. We have developed the CEVOP methodology that combines both in a single co−evolutionary algorithm.

In this work we present the behavioural model, interaction model, and the planning approach through the CEVOP methodology of the SoccerTeam prototype. Finally, we conclude with some remarks to the work. A more detailed discussion of the CEVOP methodology can be found in [5], [6].

2 The Behavioural Model

Since no other than the planning behaviour is implemented in the prototype, the intelligence of the players is restricted to the capabilities of the combined game strategies and the decision mechanism.

2.1 Planning

Planning in SoccerTeam is a two−stage process:

```
1) Plan generation through CEVOP runs
2) Plan execution in SoccerTeam simulation runs
```

Each CEVOP run returns usually several plans, all for one specific game constellation. The number of constellations, even for non−continues positions, is huge. Therefore, the planning process is iterated until the overall grade of success of SoccerTeam becomes satisfactory in simulation runs.

2.2 Objects Of The Planning Process

Following objects of the Soccer planning domain and their relationships are essential for the planning process:

- A plan usually consists of several plan steps.
- A plan step is interpreted as a primitive game strategy that promises individual success for a specific game constellation.
- A plan is interpreted as a combined game strategy, consisting of three successive primitive game strategies that promise individual success for three successive game constellations.
- A team strategy consists of one combined game strategy for each player, that can be applied for the same specific game constellations. It promises an advantages game constellation after applying the strategy.

2.3 Planning Strategies

We have designed different types of game strategies as potential parts of a plan. They are grouped into static game strategies and dynamic game strategies. The latter is further subdivided into basic game strategies, goalkeeper strategies, primitive offensive game strategies, primitive defensive game strategies, primitive game starting or continuing strategies, combined game strategies, and coach's game strategies [5]. Where, the latter are meta strategies over the others.

In the plan generation process for constructing a combined game strategy, the change of the game constellation, after all players' first steps have been applied, is successively considered in each of the remaining two plan steps. However, in the plan execution process, after all players' first steps have been applied, the resulting game constellation may not be the same as planned. Two factors cause this effect:

- Each player has restricted sensoring capabilities, which causes to view the current constellation differently and therefore plan differently.
- The opponent's planning and execution approaches may be different (heterogeneous teams).

Thus, planning three steps ahead aims at finding the most successful first plan step for the current constellation. The remaining two steps represent actually a success factor for the first step and strengthens its strategic value. After the first step of a plan has been applied and the constellation has changed, another more suitable combined game strategy may be chosen.

2.4 Optimising The Planning Process

In unsupervised learning, in iterative applications the CEVOP algorithm is initialised always with random values. In this case, the set of combined game strategies that covers all possible game constellations found is a statistical mean value. This value is usually better than that of an exhaustive search. However, the statistical mean value can further be reduced, if some heuristics are utilised, which means that learning will be supervised.

In supervised learning, the CEVOP algorithm is initialised with specific game constellations, in order to get a solution for which no successful plan exists, yet. Sample heuristics are specific game starting positions for throw−in, kick−off, or free−kick. For this purpose, related game constellations schemes have been identified and the players' initial co−ordinates randomly concentrated around the ball's co−ordinates.

2.5 Learning

Theoretically, any plan that was not known by the agent and was found by CEVOP from the set of all possible plans, represents a learning process. If the set of primitive game strategies remain the same for all CEVOP runs and the most significant game constellations have been learned, then the learning factor turns to decrease with any further CEVOP run. Meaning, that any new plan will increasingly approximate the already learned plans.

3 The Interaction Model

The interaction model enables an agent to interact with its environment. Here, we discuss the communication, co−operation, competition, and co−ordination models of SoccerTeam.

3.1 The Communication Model

The communication protocol of the Soccer simulation server [2] is utilised additionally for inter−team communication of some primitive concepts. A team member may send a message to the others, in order to transmit its intention expressed by the information below:

- The coach wants the players to apply a specific meta strategy.
- The sender wants the receiver to pass the ball to the sender.
- The sender wants the receiver to know that the sender is about to pass the ball to the receiver.

3.2 The Co−operation Model

Although, all 11 players of a team co−operate, because of the difficulty to apply the same fitness function to heterogeneous populations, we must distinguish types for co−operativ evolution depending on population diversity:

- Co−operating homogeneous players: Since, the knowledge bases are homogeneous, the plans of the 10 players can evolve in the same search space, which is represented by one population.
- Co−operating heterogeneous players: Since, the knowledge bases are heterogeneous, the plans of the 10 players evolve in a different search space than those of the goal keeper, which is represented by two populations. Besides the fitness function of each population, a fitness sharing function is defined, which represents the co−operation relationship.

3.3 The Competition Model

We represent competition by different populations, since the sought solutions are orthogonal and therefore require different fitness functions. Population diversity no more effects this situation, since we already have different populations. Besides the fitness function of each population, a fitness sharing function is defined, which represents the competition relationship.

3.4 The Co−ordination Model

The co−operation structures, discussed above, imply some means for co−ordination among the players of a team. Further control is implemented with the meta strategies, which the coach may apply depending on specific game constellations and overall match situations [5].

4 Planning With CEVOP In SoccerTeam

The principle structures, co−operation, competition, and co−ordination, found in Soccer game, have explicitly been mapped onto the principle interaction structures of the co−evolutionary algorithm. Now we present the mapping of these structures in a compound form, inside the co−evolutionary algorithm.

4.1 Populations

The homogeneous/heterogeneous and co−operation/competition structures of SoccerTeam are depicted in (*Figure 1*). The co−evolutionary algorithm will operate with this schema of populations and relationships.

In Soccer game also goal keepers compete against each other. However, it is quite unlikely that both will meet on the field. Therefore, this case is unrealistic and can be avoided in the learning phase.

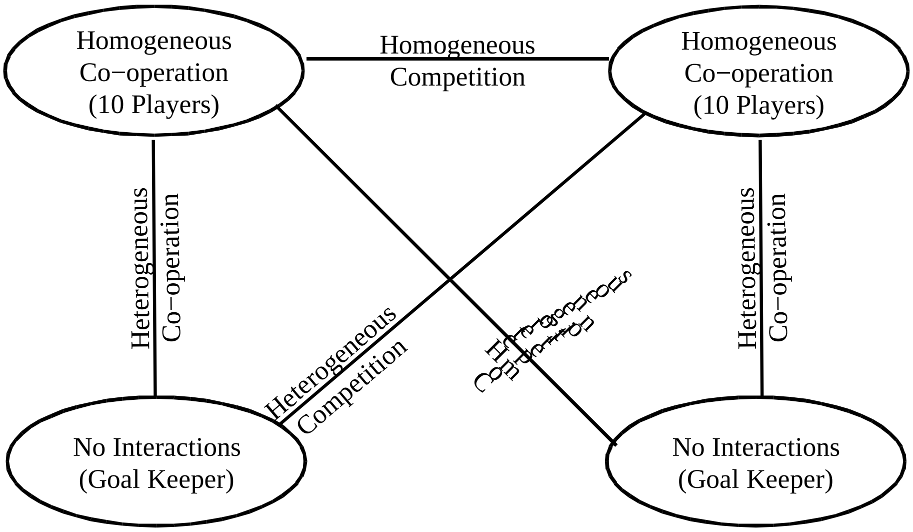

Figure 1: Co−operative/Competitive Evolution Of The Four Homogeneous/Heterogeneous Plan Populations

4.2 Phenotype Structure

A phenotype represents 11 combined game strategies $S_{p,i}$, one for each player. Where p = 1, ..., 11. Each of them consists of i = 1, 2, 3 primitive game strategies $S_{p,1}$, ..., $S_{p,3}$ with their relative locations $\Delta x_{1,1}$, $\Delta y_{1,1}$, ..., $\Delta x_{11,3}$, $\Delta y_{1,3}$, one for each player (*Table 1*). Each row of the table represents one combined game strategy. Three succeeding columns represent all first primitive game strategies of all 11 players. Thus, one phenotype represents, for a specific game constellation, a team plan solution consisting of 11 plans, one for each player. Where each plan consists further of three plan steps.

Table 1. Phenotype Structure That Represents One Team Strategy

$S_{1,1}$	$\Delta x_{1,1}$	$\Delta y_{1,1}$	$S_{1,2}$	$\Delta x_{1,2}$	$\Delta y_{1,2}$	$S_{1,3}$	$\Delta x_{1,3}$	$\Delta y_{1,3}$
	...			...			...	
$S_{11,1}$	$\Delta x_{11,1}$	$\Delta y_{11,1}$	$S_{11,2}$	$\Delta x_{11,2}$	$\Delta y_{11,2}$	$S_{11,3}$	$\Delta x_{11,3}$	$\Delta y_{11,3}$

4.3 The Generic Co−evolutionary Algorithm

Following algorithm co−evolves the n = 4 populations according their m interaction types:

```
1) For all n=4 populations
1.1) Randomly pickup plan steps from Pi for population Ai
2) For all n=4 populations
2.1) Cross−over inside Ai
3) For all n=4 populations
3.1) Mutate distance values inside Ai
4) For all n=4 populations
4.1) Calculate Fitness Fi of Population Ai
4.2) For all m interaction relationships of population Ai
4.2.1) Calculate fitness Fij for interaction of Ai with Aj (i ≠ j)
4.2.2) Fc = Fc + Fij
4.3) Fi = (Fi + Fc / m) / 2 and Fc = 0
5) IF fitness F1 AND ... AND Fm satisfied OR other termination THEN
   GO TO 7
6) For all n populations
6.1) Fitness selection on Ai
6.2) GO TO 2
7) For all n populations
7.1) PRINT Ai
```

A_i is the plan population to be optimised. P_i is the set of all possible individual plan steps of agent i. Fitness F_i of population A_i is determined by accumulating all m fitness values for interrelationships F_{ij}: $F_i = F_{i1} + F_{im}$. F_{im} represents the co−operation and/or competition strengths of agent i with the others.

4.4 Interpretation Of The Algorithm For The Case Of Homogeneous Co−operation

In case of the 10 co−operating homogeneous players of a team, the set of all possible plans P_i is identical for all agents A_{i1}, ..., A_{i10}. Thus, the plans can co−operatively evolve inside a single population. For this purpose the plans of all these players are represented inside a single phenotype.

On the other hand, the two goal keepers’ plans evolve in distinct populations, since they compete against each other and since their plans are heterogeneous to the other players.

4.5 Crossover

Uniform cross−over is applied on all genes, i.e. the primitive game strategies $S_{1,1}$, ..., $S_{13,3}$ and the distance values $\Delta x_{1,1}$, $\Delta y_{1,1}$, ..., $\Delta x_{11,3}$, $\Delta y_{1,3}$.

4.6 Mutation

Mutation is applied only on the distance values $\Delta x_{1,1}$, $\Delta y_{1,1}$, ..., $\Delta x_{11,3}$, $\Delta y_{1,3}$. Where, the range is dependent on the related primitive game strategy.

4.7 Fitness Evaluation

The definition of the fitness functions is by far the most critical part of the CEVOP methodology, since they drive the populations towards the desired fitness. For each of the interaction types homogeneous/heterogeneous co−operation/competition one fitness function is defined. Common to all is some functionality, which is further grouped depending on the strategy to be evaluated. A brief description of these groups is given below.

Before a population is evaluated by a fitness function and after cross−over and mutation has been applied, first some services routines perform some semantic checks:

- All combined game strategies are applied on the current game constellation.
- A physically impossible move is evaluated to zero.
- A move that violates the game rules is evaluated to zero.

Some fitness evaluations for offensive strategies:

- A move of the ball towards the opponent's goal is evaluated higher than a move off the opponent's goal.
- A move of the ball to a location closer to 90° angle to the goal is evaluated higher than a move to a location closer to 0° angle.

Some fitness evaluations for defensive strategies:

- If a player is the closest one from its team to the ball, then trying to get the ball is evaluated higher for that player.
- If other team members are closer to the ball than a specific player, then covering an opponent is evaluated higher for that player.

The fitness value of a combined game strategy is calculated by a suitability function that sorts all combined game strategies in decreasing suitability, in order to determine the next player O_b, to which the ball will be passed.

4.8 Overall Algorithmic Steps

The algorithmic steps of the CEVOP methodology are given below.

```
1) Apply the Co−evolutionary Algorithm for a Random/Desired Game
   Constellation
2) Store the Resulting Combined Game Strategies for Players/Goal
   Keepers in the Knowledge Base of each Player/Goal Keepers
```

```
3) IF SoccerTeam is not Successful in a Simulation Run THEN GO TO
   1
```

The number of iterations of this algorithm depends on the grade of success of the team over another team in a simulation run. However, after a specific threshold, each found combined game strategy will look increasingly similar to the previously found strategies. Thus, after a number of iterations the found set of combined game strategies will saturate with respect to its successfulness in simulation runs.

4.9 Combined Evaluated Structures

Implied by the properties plan diversity and interaction type, the structures evolve in some combinations. These combinations are recapitulated in the following:

- Each primitive game strategy and its distance values evolve combined to one player strategy.
- Three primitive game strategies with their distance values evolve to a combined game strategy for one player.
- 11 combined game strategies evolve combined to one team strategy; one for each team member.
- Two team strategies evolve combined to one solution of the current game constellation.

4.10 Improving The Results

The goal of the planning process is to find a set of combined game strategies for a team, such that they cover all possible game constellations. In other words, we are seeking for a set of combined game strategies that can provide for any constellation a successful team plan. For this purpose we inspect the co−evolutionary algorithm more detailed.

While exploring the search space opened by all possible plans, the algorithm may pass various sub−solutions that could be interesting alternative plans for some game constellations. Techniques, such as niching, crowding, fitness sharing, shared sampling, hall of frame, phantom parasite, brood selection, etc. are employed in co−evolutionary algorithms [10], [9], [8] for the purpose of finding alternative representative solutions or partial representative sub−solutions. Each of such solutions need to be proved, in how far it could contribute to the set of already found solutions, in order to improve the overall success.

5 Conclusions

We have introduced the concepts of the SoccerTeam prototype that acts as a client to the RoboCup simulation server. We have emphasised the architectural concepts knowledge base, behaviour, communication, co−operation, competition, and co−ordination. Where, for the planning process of an agent, the three latter are approached combined in the CEVOP methodology. Our goal is to find a set of representative combined game strategies, which is as small as possible and covers all significant game constellations. SoccerTeam is currently pre−registered for the RoboCup Soccer simulation league 2002.

References

1. Bull, Lawrence; Fogarty, Terence C.; 1996; "Evolutionarey Computing in Cooperative Multi−agent Environments”; *Adaptation, Coevolution and Learning in Multiagent Systems*; AAAI Press; Menlo Park

2. Chen, Mao; Foroughi, Ehsan; Heintz, Fredrik; Zahn, Xiangh Huang; Kapetanakis, Spiros; Kostiadis, Kostas; Kummeneje, Johan; Noda, Itsuki; Obst, Oliver; Riley, Pat; Steffens, Timo; Wang, Yi; Yin; Xiang; 2001; “RoboCup Soccer Server”; *RoboCup Federation*; http:// www.robocup.org/

3. Grefenstette, John; Daley, Robert; 1996; "Methods for Competitive and Cooperative Co−evolution”; *Adaptation, Coevolution and Learning in Multiagent Systems*; AAAI Press; Menlo Park

4. Matsubara, Hitoshi; Noda, Itsuki; Hiraki, Kazuo; 1996; "Learning of Cooperative actions in multi−agent systems: a case study of pass play in Soccer"; *Adaptation, Coevolution and Learning in Multiagent Systems*; AAAI Press; Menlo Park

5. Kumova, Bora I.; 2002; “The The Game Strategies of SoccerTeam”; submitted to *TAINN’02*; Ýstanbul University

6. Kumova, Bora I.; 2002; "Co−evolutionary Planning”; PhD thesis (May 2002); *Dokuz Eylul University*

7. Potter, Mitchell A.; 1997; "The Design and Analysis of a Computational Model of Cooperative Coevolution"; dissertation; *George Mason University*; Virginia

8. Rosin, Christopher Darrell; 1997; “Coevolutionary search among adversaries”; *University of California*, San Diego; ISBN 0−591−42206−9

9. Goldberg, D. E.; Richardson, J.; 1987; "Genetic algorithms with sharing for multimodal function optimization"; *Proceedings of the Second International Conference on Genetic Algorithms*

10. Mahfoud, S. W.; 1995; "Niching methods for Genetic Algorithms"; PhD thesis; *University of Illinois at Urbana−Champaign*

11. Luke, Sean; Hohn, Charls; Farris, Jonathan; Jackson, Gary; Hendler, James; 1997; “Co−Evolving Soccer Softbot Team Coordination with Genetic Programming”; *IJCAI’97*; Springer

12. Haynes, D. Thomas; Sen, Sandip; 1997; “Co−adaptation in a Team”; *IJCIO*;