UvA Trilearn 2004 Team Description

Jelle R. Kok, Nikos Vlassis, and Frans Groen

Faculty of Science, University of Amsterdam
{jellekok,vlassis,groen}@science.uva.nl

Abstract. This paper shortly describes the main features of the *UvA Trilearn* soccer simulation team, which participates since 2001 in the RoboCup competition. The main concepts of the previous teams are addressed, followed by the improvements introduced in *UvA Trilearn 2004*. The latter are related to the coordination of the agents using the coordination graph structure introduced in *UvA Trilearn 2003*. We will show how we apply this model both for action selection and for teammate modeling.

1 Introduction

The *UvA Trilearn 2001* soccer simulation team [1] was built by two masters' students for their graduation project. Much of the effort in this team had gone into getting the lower levels to work, since we felt that these would be the most crucial for the success of the team. This has among other things led to a multi-threaded three-layer architecture with an advanced synchronization method, a probabilistic world model from which several high-level conclusions could be derived and a layered skills hierarchy.

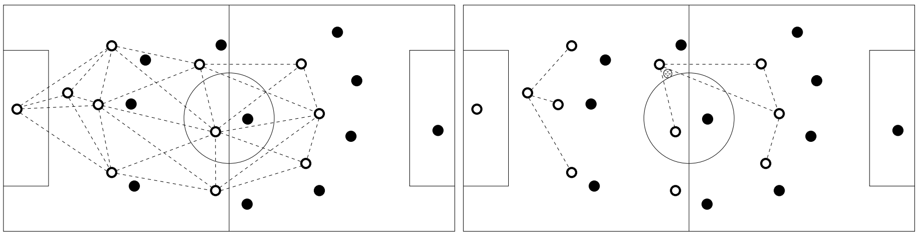
UvA Trilearn 2002 [4] contained several improvements including improved localization methods using particle filters, behavior modeling of teammates, and an action selection method based on a priority-confidence model.

In *UvA Trilearn 2003*, we improved the intercept skill by taking opponents into account to determine the best interception point and improved the passing behavior of the players by taking much more passing options in consideration. This was possible by applying an efficient algorithm [8] to calculate the interception time of an opponent. However, the most important improvement was the introduction of coordination graphs [3] in order to coordinate the actions of the agents. With these improvements, *UvA Trilearn 2003* won the German Open, American Open and the RoboCup-2003 world championships.

We released a large part of our *UvA Trilearn 2003* source code[1] on which several other teams have based their work. The released code consists of our lower levels (synchronization, world model, basic agent skills) in combination with a simple high-level strategy, similar to the one released by FC Portugal [6] after RoboCup-2000 to make a working team.

In this paper, we shortly explain how we apply the coordination graph model to coordinate the different agents and to model the behavior of our teammates.

[1] Available from `http://www.science.uva.nl/~jellekok/robocup/`.



(a) Complete coordination Graph.

(b) Possible coordination graph after role distribution.

Fig. 1. After observing the ball position and knowing the ball position, the coordination graph is simplified considerably.

2 Coordination Graphs

In systems where multiple agents have to coordinate their actions, it is infeasible to model all possible joint actions since this number grows exponentially with the number of agents. Fortunately, most problems exhibit the property that each agent only has to coordinate with a small subset of the other agents, e.g., a soccer agent controlling the ball only has to coordinate with agents within passing distance. A principled approach to exploit such dependencies involves the use of a coordination graph (CG), which represents the coordination requirements of a system [2].

In this graph, each node represents an agent, and an edge indicates that the corresponding agents have to coordinate their actions. In order to reach a jointly optimal action, a variable elimination algorithm is applied that iteratively solves the local coordination problems one by one and propagates the result through the graph using a message passing scheme. In a context-specific CG [3] the topology of the graph is first dynamically updated based on the current state of the world before the elimination algorithm is applied. See [3, 5] for details of the algorithm.

The continuous nature of the state space makes the direct application of context-specific CGs difficult. Therefore, we appropriately 'discretize' the continuous state by assigning *roles* to the agents and then, instead of coordinating the different agents, coordinate the different roles [7]. It turns out that such an approach offers additional benefits: the set of roles not only allows for the definition of natural coordination rules that exploit prior knowledge about the domain, but also constrain the feasible action space of the agents. This greatly simplifies the modeling and the solution of the problem at hand. Fig. 1 shows how the distribution of roles after observing the ball position simplifies the coordination graph considerably.

In our *UvA Trilearn 2003* team, we used this framework to improve upon the passing between the teammates. Instead of being reactive (a player only starts intercepting after it observes a change in the ball velocity), the coordination framework makes sure that the agents already move to the free space the ball will be passed to before the pass is actually given. An example of a rule to specify such a coordination pass can be defined as follows:

$$\begin{aligned}\langle p_1^{passer} \quad ; \quad & \text{has-role-receiver}(j) \quad \wedge \\ & \neg\text{isPassBlocked}(i, j, dir) \quad \wedge \\ & a_i = \text{passTo}(j, dir) \quad \wedge \\ & a_j = \text{moveTo}(dir) : u(j, dir)\rangle \ \forall j \neq i\end{aligned}$$

where agent i is the passer, agent j is a possible receiver and dir is the direction relative to agent j to which the pass is oriented. More rules are added for each role to specify other possible actions, e.g., dribbling, moving to a strategic position, etc. Using the variable elimination algorithm the individual actions that maximize the received payoff for all applicable rules in the current context are determined. See [5] for details about the used rules and the algorithm.

The original algorithm depends on a message passing system in order to distribute the rules over the network. However, in the RoboCup simulation domain it is not possible to perform this amount of communication. However, the variable elimination algorithm can still be applied if we further impose the requirement that the payoff function of an agent i is common knowledge among all agents that are *reachable* from i in the CG. Since only agents that are reachable in the CG need to coordinate their actions, the above requirement in fact frees agents from having to communicate their local payoff functions during optimization. These common knowledge assumption about the fully observable state cannot be made during competition since every agent only receives information of the part of the field to which its neck is oriented. However, the coordination graph structure specifies which parts of the state are relevant for coordination, i.e., the neighbors in the graph and their associated state variables. Therefore, we adjust the looking mechanism of the agents to actively orient their neck to the part of the field in which its neighbors in the graph are located and then assume that this part of the world is common knowledge among these agents. When all involved agents observe this information they can independently solve the local graph which is disconnected from the rest of the CG and so compensate for the missing state information. In our example, the passer and the receivers thus change their looking direction to their neighbors in the graph in order to get a good approximation of the relevant part of the state needed for coordination and are not interested in the passive players which are not connected to their subgraph.

In our *UvA Trilearn 2004* team we use this approach to improve upon the information that is needed to apply the variable elimination algorithm. Additionally, we use the calculated joint action to update the information of each teammate with its calculated individual action in order to improve upon the estimated locations of the teammates. Finally, we are currently working on extending the coordination dependencies between the agents, e.g., to coordinate the defenders and to coordinate marking the opponent players to obstruct their play when they are in control of the ball.

3 Conclusion

In this paper we briefly addressed the main contributions of our simulation team *UvA Trilearn*. The main improvement in *UvA Trilearn 2004* is the specification of additional coordination requirements between the players using the coordination graph structure introduced in *UvA Trilearn 2003* and the modeling of teammates using their calculated action derived from the variable elimination algorithm.

Acknowledgements

This research is supported by PROGRESS, the embedded systems research program of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the Technology Foundation STW, project AES 5414.

References

1. Remco de Boer and Jelle R. Kok. The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team. Master's thesis, University of Amsterdam, The Netherlands, February 2002.
2. C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems 14*, pages 1523–1530. The MIT Press, 2002.
3. C. Guestrin, S. Venkataraman, and D. Koller. Context-specific multiagent coordination and planning with factored MDPs. In *Proc. 8th Nation. Conf. on Artificial Intelligence*, pages 253–259, Edmonton, Canada, July 2002.
4. Jelle R. Kok, Remco de Boer, Nikos Vlassis, and F.C.A. Groen. UvA Trilearn 2002 team description. In G. Kaminka, P.U. Lima, and R. Rojas, editors, *RoboCup 2002: Robot Soccer World Cup VI*, page 549, Fukuoka, Japan, 2002. Springer-Verlag.
5. Jelle R. Kok, Matthijs T. J. Spaan, and Nikos Vlassis. Multi-robot decision making using coordination graphs. In A.T. de Almeida and U. Nunes, editors, *Proceedings of the 11th International Conference on Advanced Robotics, ICAR'03*, pages 1124–1129, Coimbra, Portugal, June 30-July 3 2003.
6. Luís Paulo Reis and José Nuno Lau. FC Portugal Team Description: RoboCup-2000 Simulation League Champion. In Peter Stone, Tucker Balch, and Gerhard Kraetszchmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 29–40. Springer Verlag, Berlin, 2001.
7. M. T. J. Spaan, N. Vlassis, and F. C. A. Groen. High level coordination of agents based on multiagent Markov decision processes with roles. In A. Saffiotti, editor, *IROS'02 Workshop on Cooperative Robotics*, pages 66–73, Lausanne, Switzerland, October 2002.
8. Peter Stone and David McAllester. An Architecture for Action Selection in Robotic Soccer. In *Fifth International Conference on Autonomous Agents*, pages 316–323. ACM Press, 2001.