UvA Trilearn 2005 Team Description

Jelle R. Kok and Nikos Vlassis

Informatics Institute, University of Amsterdam, The Netherlands
{jellekok,vlassis}@science.uva.nl

Abstract. This paper shortly describes the main features of the *UvA Trilearn* soccer simulation team, which participates in the RoboCup competition since 2001. In the last years we mostly concentrated on the coordination of the different agents. For this we applied the framework of coordination graphs. In this framework, a high-level strategy is specified using value rules which describe the effectiveness of a (possible joint) action in a specific situation. During a game, a variable elimination algorithm is applied on all applicable rules in order to find the individual actions for the agents which maximize the global effectiveness. Variable elimination is exact, but it does not scale well to systems where many agents depend on each other. In our UvA Trilearn 2005 team, we therefore experiment with the usage of the max-plus algorithm, as an approximate alternative to variable elimination.

1 Introduction

UvA Trilearn started as a masters' project in 2001. In the first years, much of the effort went into building a good base code. This resulted in a multi-threaded three-layer architecture with an advanced synchronization method, a probabilistic world model, and a set of basis agent skills. On top of this, we first built a simple high-level team strategy, similar to the one released by FC Portugal [1], to make a working team. This base code, which we used as a basis for our team that won the World Championship in RoboCup-2003, is used by several teams and publicly available[1].

In the last few years, we mainly concentrated on coordinating the different agents. For this we applied the framework of coordination graphs [2]. As part of this framework, a variable elimination algorithm is used. Although this method is exact, it is not appropriate for real-time systems. For this year's team, we investigate the max-plus algorithm, an instance of the belief propagation algorithm, as an approximate alternative to variable elimination.

The setup of this paper is as follows. In Section Section 2 we will review the coordination graph framework using variable elimination. In Section Section 3 we will describe the max-plus algorithm and in Section 4 we will end with a conclusion.

[1] It can be downloaded from `http://www.science.uva.nl/~jellekok/robocup/`.

2 Coordination Graphs

In order to coordinate the different players on the field, we apply the framework of coordination graphs [2]. We (manually) specify a complete (high-level) team strategy using value rules, which are propositional rules over state and action variables. Each value rule is associated with a local payoff that is obtained when the corresponding state and action variables are applicable. The sum of the payoffs of the applicable rules define the global contribution to the system. Such a set of value rules can be regarded as a sparse representation of the complete state-action representation, since it decomposes the global coordination problem into the sum of a set of smaller local terms. An example of a value rule representing a coordinated pass between player 1 and 2 that is applicable when this pass is not blocked by an opponent and contributes a local payoff of 10 looks as

$$\langle \rho_1; \neg\text{passBlocked(1,2)} \wedge a_1 = \text{passTo(2)} \wedge a_2 = \text{receivePass(1)} : 10 \rangle,$$

while an individual dribble action could be specified as

$$\langle \rho_2; \text{isEmptySpace(1,north)} \wedge a_1 = \text{dribble(north)} : 5 \rangle.$$

Given a set of value rules, a coordination graph can be constructed. Each node represents an agent, while an edge indicates that the corresponding agents have to coordinate their actions (and both agents are thus involved in at least one value rule). In order to determine a joint action for a specific situation maximizing the sum of the payoffs of the applicable rules, the following procedure is applied. First all rules that are not applicable in the current state are removed. In most cases, this simplifies the topology of the graph significantly. In order to reach a jointly optimal action, a variable elimination (VE) algorithm is then applied that iteratively eliminates an agent by solving a local coordination problem and propagating the result to its neighbors in the graph using a message passing scheme. When all but one agent are eliminated, this last agent fixes its strategy and a pass in the reverse order is performed in which each agent fixes its strategy by conditioning on the strategies of its neighbors in the graph. This method is exact and will always return the joint action that maximizes the sum of the payoff values in the applicable rules. See [3] for details of the usage of this method in our RoboCup simulation team.

However, for highly connected network this method scales exponentially in the number of agents. Furthermore, VE will only produce its final result after the complete algorithm is finalized. This makes is not always appropriate for real-time multiagent systems where decision making must be done under time constraints. For this reason, we investigated the use of the max-plus algorithm, analogous to the belief propagation algorithm [4, 5] in Bayesian networks, as an alternative to VE for multiagent decision making. We will discuss this method in more detail in the next section.

3 Max-plus Algorithm

In this Section, we will briefly review the max-plus algorithm. See [10] for a more detailed description,

The *max-plus algorithm* [4–7] is a popular method for computing the *maximum a posteriori* (MAP) configuration in an (unnormalized) undirected graphical model. This method, analogous to the belief propagation (BP) or sum-product algorithm, operates by iteratively sending messages $\mu_{ij}(a_j)$ between agent i and j over the edges of the graph. We can apply this procedure to our multiagent decision making problem. Suppose that we have a coordination graph G which is constructed from all applicable value rules in the current situation. Each node in the graph represent an agent, while the edges define the dependencies between the agents. The global payoff for a given situation, can then be defined as the sum of all local rules

$$u(a) = \sum_{i \in V} f_i(a_i) + \sum_{(i,j) \in E} f_{ij}(a_i, a_j) \tag{1}$$

Here f_i denotes a value rule based on an individual action a_i. Furthermore, (i, j) denotes a pair of neighboring agents (an edge in G), and f_{ij} is a value rule that involves a pair of actions (a_i, a_j) to a real number $f_{ij}(a_i, a_j)$.

Again, the goal is to find the optimal joint action $a^* = \arg\max_a u(a)$ that maximizes (1). For this, each agent i (node in G) repeatedly sends a message μ_{ij} to one of its neighbors $j \in \Gamma(i)$, where μ_{ij} is a local payoff function that maps an action a_j of agent j to a real number $\mu_{ij}(a_j)$ and is defined as follows:

$$\mu_{ij}(a_j) = \max_{a_i} \Big\{ f_i(a_i) + f_{ij}(a_i, a_j) + \sum_{k \in \Gamma(i) \setminus j} \mu_{ki}(a_i) \Big\} + C_{ij}, \tag{2}$$

where the notation $\Gamma(i) \setminus j$ means all neighbors of node i except node j and C_{ij} is an arbitrary constant (that does not depend on a_j). This message can be understood as the approximately maximum payoff agent i can produce given the action of agent j and equals the sum of the payoff functions f_i and f_{ij} and all incoming messages to agent i except that from agent j. Fig. 1 shows a graphical representation of such messages. The agents keep exchanging messages until they converge (do not change anymore) or a certain amount of time has elapsed.

It can be shown that when G is cycle-free, max-plus will converge to a fixed-point within a finite number of steps [4, 5]. Then if we define

$$g_i(a_i) = f_i(a_i) + \sum_{j \in \Gamma(i)} \mu_{ji}(a_i), \tag{3}$$

we can show that

$$g_i(a_i) = \max_{\{a' | a'_i = a_i\}} u(a') \tag{4}$$

holds. Each agent i can now individually select its locally optimal action $a_i^* = \arg\max_{a_i} g_i(a_i)$. If this action is unique for every agent i (there is only one

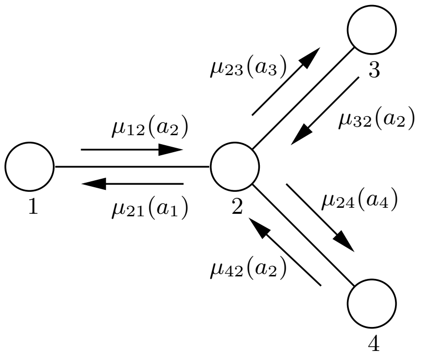

Fig. 1. Graphical representation of different messages μ_{ij} in a graph with four agents.

maximizing action), the globally optimal joint action $a^* = \arg\max_a u(a)$ is also unique and has elements $a^* = (a_i^*)$. Note that this global optimal joint action is computed by only local optimizations (each node maximizes $g_i(a_i)$ separately).

In graphs with cycles, these convergence proofs do not hold, and there are no guarantees that either max-plus will converge or that the local maximizers $a_i^* = \arg\max_{a_i} g_i(a_i)$ will correspond to the global optimum. In [5] it is shown that a fixed point of message passing exists in graphs with cycles, but there is no known algorithm yet that can provably converge to such a solution. However, in practice, the max-product algorithm has been successfully applied within graphs with cycles [6, 8, 9], which also makes it an attractive method in our setting. Even when the messages do not converge, the corresponding joint action shows near optimal results. See [10] for details and our experiments on randomly generated graphs.

Finally, note that the agents in the 2D simulator are not allowed to directly communicate with each other. Therefore, in order to implement the max-plus algorithm in our RoboCup simulation team, we have to model the complete procedure separately in each agent using some common knowledge assumptions. See [3] for a description of such an approach when variable elimination is used to compute the joint action, which has to cope with exactly the same problem.

4 Conclusion

In this paper we addressed the main contributions of our simulation team *UvA Trilearn*. The main improvement in *UvA Trilearn 2005* is the usage of the max-plus algorithm in order to speed-up the computation of determining the joint action in the coordination graph framework introduced in *UvA Trilearn 2003*.

The max-plus algorithm computes the individual actions significantly faster than the previous used variable elimination algorithm, while it still results in comparable results. Therefore, we can apply the coordination graph framework on a much larger and more complicated set of value rules as in the previous years.

We are currently incorporating the max-plus algorithm in our *UvA Trilearn 2005* simulation team and are also extending the team strategy to include more specific and fine-grained coordination.

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. Reis, L.P., Lau, J.N.: FC Portugal Team Description: RoboCup-2000 Simulation League Champion. In Stone, P., Balch, T., Kraetszchmar, G., eds.: RoboCup-2000: Robot Soccer World Cup IV. Springer Verlag, Berlin (2001) 29–40
2. Guestrin, C., Venkataraman, S., Koller, D.: Context-specific multiagent coordination and planning with factored MDPs. In: Proc. 8th Nation. Conf. on Artificial Intelligence, Edmonton, Canada (2002)
3. Kok, J.R., Spaan, M.T.J., Vlassis, N.: Non-communicative multi-robot coordination in dynamic environments. Robotics and Autonomous Systems **50** (2005) 99–114
4. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufman, San Mateo (1988)
5. Wainwright, M., Jaakkola, T., Willsky, A.: Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. Statistics and Computing **14** (2004) 143–166
6. Yedidia, J., Freeman, W., Weiss, Y.: Understanding belief propagation and its generalizations. In: Exploring Artificial Intelligence in the New Millennium. Morgan Kaufmann Publishers Inc. (2003) 239–269
7. Vlassis, N., Elhorst, R., Kok, J.R.: Anytime algorithms for multiagent decision making using coordination graphs. In: Proc. of the International Conference on Systems, Man and Cybernetics, The Hague, The Netherlands (2004)
8. Crick, C., Pfeffer, A.: Loopy belief propagation as a basis for communication in sensor networks. In: Proceedings of the 19th Conference on Uncertainty in AI. (2003)
9. Murphy, K., Weiss, Y., Jordan, M.: Loopy belief propagation for approximate inference: An empirical study. In: Proc. 15th Conf. on Uncertainty in Artificial Intelligence, Stockholm, Sweden (1999)
10. Kok, J.R., Vlassis, N.: Using the max-plus algorithm for multiagent decision making in coordination graphs. In: RoboCup 2005: Robot Soccer World Cup IX, Osaka, Japan (2005) to appear.