USING THE MAX-PLUS ALGORITHM FOR MULTIAGENT DECISION MAKING IN COORDINATION GRAPHS[1]

Jelle R. Kok Nikos Vlassis

Informatics Institute, University of Amsterdam,
Kruislaan 403, 1098SJ, Amsterdam, The Netherlands

The full version of this paper [2] appeared in the Proceedings of the 9^{th} RoboCup International Symposium in Osaka, Japan, July 2005 and won the best scientific paper award.

1 Introduction

A multiagent system (MAS) consists of a group of agents that interact which each other [4]. In such systems agents act individually, but the outcome of their actions might differ based on the behavior of the other agents. In this paper, we focus on a cooperative MAS problem in which each agent selects an individual action a_i and the resulting joint action $a = (a_i)$ should maximize the global payoff function $u(a)$.

Fortunately, many MAS problems exhibit the property that each agent only depends on a small subset of the other agents, e.g., those who are spatially close. Coordination graphs (CG) [1] is a framework for multiagent coordination in which $u(a)$ is decomposed into a linear combination of local payoff functions, each involving only a few agents. In order to determine the optimal joint action a that maximizes $u(a)$, a variable elimination (VE) algorithm was proposed in [1]. This method operates by iteratively eliminating an agent after performing a local maximization step that involves an enumeration of all possible action combinations of its neighbors. Although VE is exact and always produces the optimal joint action, it scales exponentially in the number of agents for densely connected graphs. Moreover, it is not appropriate for real-time systems as it requires that the complete algorithm terminates before a solution can be reported. In this paper we describe the max-plus algorithm as an approximate alternative to VE.

2 Max-Plus Algorithm

The *max-plus algorithm* [3], analogous to the belief propagation (BP) or sum-product algorithm in Bayesian networks, is a method for computing the *maximum a posteriori* (MAP) configuration in an undirected graphical model. We apply max-plus as an approximate alternative to VE for multiagent decision making. Suppose that we have a coordination graph $G = (V, E)$ with $|V|$ vertices and $|E|$ edges. The global payoff function $u(a)$ can be decomposed as

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

Here $f_i(a_i)$ denotes the payoff for the action a_i of agent i and f_{ij} is a payoff function that maps the actions (a_i, a_j) of two neighboring agents $(i, j) \in E$ to a real number $f_{ij}(a_i, a_j)$. The goal is to find the optimal joint action a^* that maximizes (1).

Each agent i repeatedly sends a message μ_{ij} to its neighbors $j \in \Gamma(i)$ where μ_{ij} maps an action a_j of agent j to a real number as follows:

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

[1] Appeared in the Proceedings of the 17th Belgian-Dutch Conference on Artifical Intelligence (BNAIC'05).

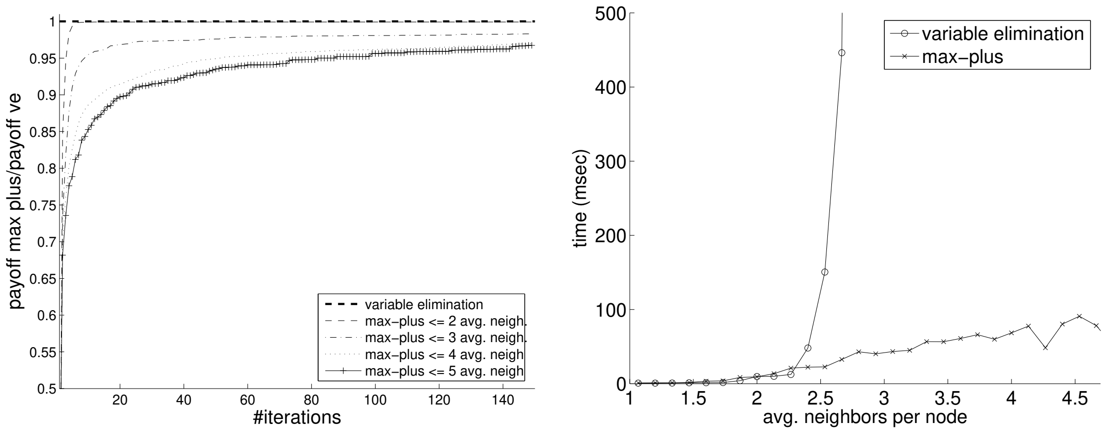

Figure 1: Payoff and timing results for graph with 15 agents and 10 actions.

in wich $\Gamma(i) \setminus j$ are all neighbors of agent i except agent j and c_{ij} is a normalization vector. This message can be understood as an approximation of the maximum payoff agent i can achieve for every action of agent j, and is computed by maximizing (over the actions of agent i) the sum of the payoff functions f_i and f_{ij} and all incoming messages to agent i except that from agent j. Messages are exchanged until they converge. If, at convergence, we define $g_i(a_i) = f_i(a_i) + \sum_{j \in \Gamma(i)} \mu_{ji}(a_i)$, we can show that $g_i(a_i) = \max_{\{a' | a'_i = a_i\}} u(a')$. Each agent i now selects its optimal action $a_i^* = \arg\max_{a_i} g_i(a_i)$. If there is one maximizing action for every agent i, the global optimal joint action $a^* = \arg\max_a u(a)$ is 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 case the local maximizers are not unique, an optimal joint action can be computed by a dynamic programming technique [5, sec. 3.1].

3 Results

Max-plus converges to the optimal solution for trees. Unfortunately, no convergence guarantees exist for graphs with cycles, although it has been successfully applied in such settings. This is also shown in Fig. 1, which shows the average results for our anytime max-plus algorithm, in which we report the best action found so far, on 3000 generated graphs with 15 agents and an increasing number of edges. Payoffs are generated using a normal distribution $\mathcal{N}(0, 1)$.

4 Conclusions

We presented the max-plus algorithm as an approximate alternative to VE. It can be implemented fully distributed and outperforms VE for densely connected graphs. These features make it an appropriate action selection technique for real-time cooperative systems.

References.

[1] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems 14*. The MIT Press, 2002.

[2] Jelle R. Kok and Nikos Vlassis. Using the max-plus algorithm for multiagent decision making in coordination graphs. In *RoboCup 2005: Robot Soccer World Cup IX*, Osaka, Japan, July 2005.

[3] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, San Mateo, 1988.

[4] N. Vlassis. A concise introduction to multiagent systems and distributed AI. Informatics Institute, University of Amsterdam, September 2003. http://www.science.uva.nl/˜vlassis/cimasdai.

[5] M. Wainwright, T. Jaakkola, and A. Willsky. Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. *Statistics and Computing*, 14:143–166, April 2004.