Efficient Multi-Agent Coordination Using Resource-Aware Junction Trees

(Extended Abstract)

N. Stefanovitch
LIP6 - UPMC
75016 Paris, France
stefanovitch@poleia.lip6.fr

A. Farinelli
University of Verona
Verona, I-37134, Italy
alessandro.farinelli@univr.it

A. Rogers, N. R.Jennings
University of Southampton
Southampton, SO17 1BJ, UK
{acr,nrj}@ecs.soton.ac.uk

ABSTRACT

In this paper we address eff cient decentralised coordination for cooperative multi-agent systems (framed as DCOPs) by taking into account the communication and computational resources of the system. We focus on techniques that exploit structural independence among agents' actions to provide optimal solutions to the coordination problem, and, in particular, we use the Generalized Distributed Law (GDL) algorithm. In this settings, we propose a novel resource aware heuristic to build junction trees and to schedule GDL computations across the agents. Our approach aims at minimising directly the total running time of the coordination process, rather than the theoretical complexity of the computation, by considering computational and communication capabilities of agents.

Categories and Subject Descriptors

I.2.11 [Computing Methodologies]: Artif cial Intelligence - Distributed Artif cial Intelligence

General Terms

Algorithms, Performance, Experimentation

Keywords

multiagent coordination, junction tree, variable elimination, heuristic algorithm, GDL, DCOP

1. INTRODUCTION

Many practical applications require the development of effective decentralised coordination techniques for cooperative multi-agent systems. Distributed Constraint Optimisation techniques have been widely used to address decentralised coordination as they are able to exploit structural independence among agents' actions, thus providing eff cient algorithms. However, these algorithms take no account of the heterogeneous computational and communication capacities available to the different agents within the system. Specifically, in many settings, it may be benef cial to delegate computations such that agents with greater than average computational capabilities can be taken advantage of and that communication between agents with poor communication links is minimised.

Cite as: Eff cient Multi-Agent Coordination Using Resource-Aware Junction Trees (Extended Abstract), N. Stefanovitch, A. Farinelli, A. Rogers and N. R. Jennings, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp.. . 1413-1414
Copyright © 2010, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

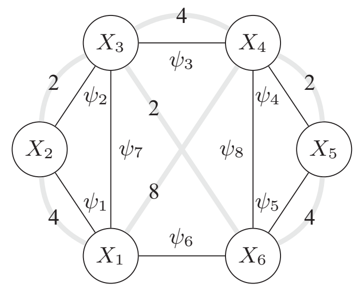

Figure 1: Example of a MAS coordination problem (communication and constraint network)

More in detail, here we focus on the Generalized Distributive Law (GDL) framework [1]. GDL works by passing messages on top of the constraint network of the DCOP and is able to provide optimal solutions when this constraint network is acyclic. However, to guarantee optimal solutions, cyclic constraint networks have to be transformed in a special graphical structure called a junction tree: an acyclic graph whose nodes (called cliques) and edges (called separators) are clusters of variables (see [3] for details). GDL messages are then computed at the level of cliques by summarising the information recursively obtained by the neighbouring cliques. Messages are transmitted back along the tree and eventually reach the root node, where the global optimum can be computed.

Against this background, our aim here is to minimise the total running time of the GDL algorithm, by exploiting knowledge about the MAS's characteristics, such as agents' processors' speeds, link bandwidth and communication network topology. Specif cally, we model the GDL solution process as a task assignment problem, where tasks represent computations that agents need to perform according to the GDL technique. Our aim is then to form a junction tree and allocate computations such that the makespan of our task assignment problem is minimised. However, optimally scheduling such a set of tasks onto a set of heterogeneous processors, even without considering communication, is known to be NP-hard [4]. Thus, in this paper, we propose an effective heuristic that works well in practise. More in details, we propose a new heuristic approach, based on the variable elimination algorithm [2], that minimises the makespan and concurrently builds and schedules GDL computations over the MAS. We describe our proposal in section 2 and present in section 3 an empirical evaluation of our approach.

Figure 1 shows an instance of a constraint network associated with a restricted communication network. The nodes represent decision variables, and the thin black edges are constraint dependencies between the variables. Constraints that hold between variables are associated to constraint functions ψ_i as shown in the f gure. The thick grey edges represent the underlying communication network between the agents responsible for the variables and the numbers on these edges correspond to the bandwidth of the communication link. The fact that poor communication links exist between the left and right part of the network should be taken into account when building the junction tree, and this is precisely the aim of our approach.

2. RESOURCE-AWARE JUNCTION TREES

In order to take into account the MAS resources while building a junction tree, we propose a novel algorithm that we call Resource Aware Heuristic (RAH), which extends the standard variable elimination algorithm. Our approach addresses at the same time two main issues: (i) f nd a suitable junction tree decomposition of the problem, by def ning cliques and allocation of constraint functions to cliques; (ii) allocate the resulting computations to agents to minimise the makespan. Specif cally, RAH def nes a new criterion to minimise when building junction trees, which is the impact on the makespan for the elimination of a given variable. In order to minimise this criterion we consider, for a given agent, the summation of three metrics: the time to process the clique on this agent, the time to transfer the set of dependant constraint functions to this agent and f nally the time to transfer the set of messages of all the dependant separators. The value returned by this heuristic is taken to be the minimum of the evaluation for all the different agents in the system. This procedure is heuristic in the way it assesses the transfer time as (i) it ignores congestion issues and (ii) cliques are created and scheduled as soon as possible, therefore the information on the set of dependant messages is only partial. However, it performs well in practice as we show in the next section.

3. EMPIRICAL EVALUATION

We empirically evaluate the RAH algorithm against DPOP, a closely related state of the art distributed constraint optimisation algorithm, and against a centralised benchmarking approach. The DFS tree of DPOP was generated using the MCN heuristic[1]. The centralised approach, which we call MS, uses a near-optimal junction tree generated with the standard variable elimination algorithm and the *minimum size* heuristic, a well known heuristic that seeks to minimise the size of the largest clique. Furthermore, in this case, all the cliques and constraint functions are allocated to the fastest agent of the system. We benchmarked those algorithms on a set of three scenarios with different constraint network structures. We consider two such structures: *ring* and *tree* with 30 and 40 agents respectively; agents are located in a square of f xed size (of 1000 unit). For each scenario we perform a set of experiments by varying the communication range (i.e., the distance within which two agents can communicate) and measure the makespan on a distributed system simulator. Results for the ring and tree instances are presented in Figure 2.

Larger communication range means higher density of the communication graph, yielding in the limit no communication constraints. It can be seen in Figure 2 that when the communication resource is scarce, RAH performs up to three times better than DPOP. As the communication range increases, the MS performance increases and it eventually outperforms RAH. Moreover, when the constraint and communication networks matches (i.e. in *tree*), the difference between RAH and DPOP is tight. Conversely it is signif cant for *ring* with RAH clearly outperforming DPOP.

[1]Maximum Connected Node, builds eff cient pseudo-trees for DPOP

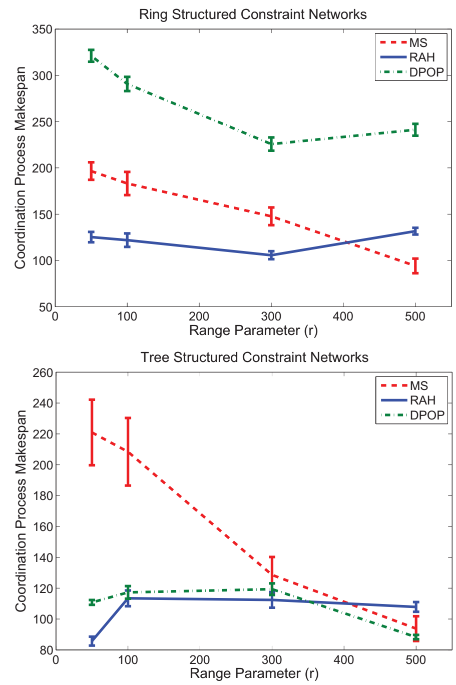

Figure 2: Coordination process makespan for the ***ring*** and ***tree*** structured constraint graph instance

4. CONCLUSION

In this work we take a f rst important step to explicitly consider MAS specif c issues (such as heterogeneity of computation and communication across the agents) when applying GDL to MAS coordination. Specif cally, we focus on the use of GDL for optimally solving DCOPs, and propose a novel approach (the RAH algorithm) to build junction trees that minimise the running time of the coordination process. Our empirical evaluation on benchmarking coordination problems shows that, when resources are scarce, our approach outperforms state of the art approaches for DCOPs.

5. REFERENCES

[1] S. Aji and R. McEliece. The generalized distributive law. *Information Theory, IEEE Transactions on*, 46(2):325–343, 2000.

[2] R. Dechter and J. Pearl. Network-based heuristics for constraint-satisfaction problems. *Artif. Intell.*, 34(1):1–38, 1987.

[3] K. Kask, R. Dechter, J. Larrosa, and A. Dechter. Unifying tree decompositions for reasoning in graphical models. *Artif. Intell.*, 166(1-2):165–193, 2005.

[4] H. Topcuoglu, S. Hariri, and M.-Y. Wu. Task scheduling algorithms for heterogeneous processors. *Heterogeneous Computing Workshop*, 0:3, 1999.