DCOPolis: A Framework for Simulating and Deploying Distributed Constraint Reasoning Algorithms

(Demo Paper)

Evan A. Sultanik Robert N. Lass William C. Regli

Department of Computer Science
Drexel University
Philadelphia, PA, USA
{eas28,urlass,wregli}@cs.drexel.edu

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; I.6.3 [**Simulation and Modeling**]: Applications

Keywords

Distributed constraint reasoning, Simulation, Middleware

1. INTRODUCTION

The proliferation of mobile computers—such as laptops, personal digital assistants, and smart phones—has propelled distributed computing into mainstream society. Over the past decade these technologies have spurred interest in both decentralized multiagent systems and wireless mobile ad-hoc networks. Such networks, however, present many challenges to information sharing and coordination. Interference, obstacles, and other environmental effects conspire with power- and processing-limited hardware to impose a number of challenging networking characteristics. Messages are routinely lost or delayed, connections may be only sporadically available, and network transfer capacity is nowhere near that available on modern wired networks. It is therefore imperative to emphasize *local* decision making and autonomy over a centralized analogue, insofar as it is possible. The majority of such decentralized decision making can be seen as a fundamental problem of propagating and then solving systems of constraints, otherwise known as Distributed Constraint Reasoning (DCR).

A large class of multiagent coordination and distributed resource allocation problems can be modeled through DCR. DCR has generated a lot of interest in the constraint programming community and a number of correct and complete algorithms have been developed to solve DCR optimization problems [6, 4, 7, 1]. Evaluating the performance of these algorithms under realistic scenarios is an important and active area of research [5, 2, 9], however, comparison is complicated by the fact that they are currently implemented in simulation; there is no record in the literature of any significant comparison of DCR algorithms on live networks. Solely evaluating distributed systems in simulation can have a number of undesirable side effects, not in the least due to differences between the individual simulators. Most of these side effects are not unique to DCR algorithms [3]. Ideally, an independent, community-endorsed framework should exist in which a single implementation of a DCR algorithm can be run both in simulation *and* deployed on a live network without modification to the underlying code [3]. This is a demonstration of the first such framework, called DCOPolis [8].

DCOPolis is an open-source[1] framework designed to abstract algorithm implementation from the underlying platform (*i.e.* hardware, network, operating system). This allows a single implementation of an algorithm to be run in simulation (*i.e.* on top of the NS2 network simulator with AgentJ), on a live ad-hoc wireless network, or even on a JavaME-enabled mobile phone. DCOPolis differs from existing DCR frameworks and simulators, however, in three fundamental ways:

1. DCOPolis was designed to allow for both simulation on a single computer (see Figure 1) and full deployment of DCR solvers on many types of live networks, including traditional wired networks and ad-hoc wireless networks;

2. DCOPolis is able to instantiate a DCR problem and start the solution process completely distributedly. This means that there is no need for configuration files, nor is there any need for a central agent/server that initializes/instantiates the rest of the group; and

3. DCOPolis supports a novel type of simulation in which the runtime of any distributed algorithm can be accurately estimated on a single physical computer. This is described in detail in [8] and visualized in Figure 2.

2. TECHNICAL CONTENT

There are two components to this demonstration: first, the algorithms and platforms currently implemented within

Cite as: DCOPolis: A Framework for Simulating and Deploying DCR Algorithms (Demo Paper), E. Sultanik, R. Lass and W. Regli, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16.,2008,Estoril,Portugal,pp.1667-1668.

[1] `http://dcopolis.sourceforge.net/`

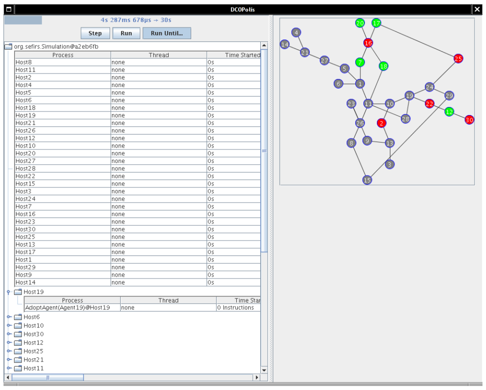

Figure 1: DCOPolis solving a distributed graph coloring problem in simulation using the Adopt algorithm.

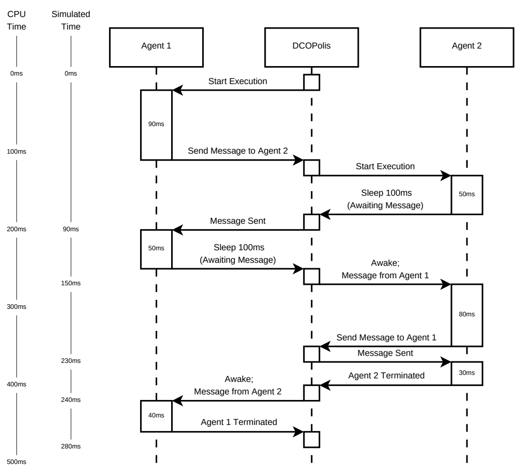

Figure 2: Sequence diagram of an execution of DCOPolis' simulated time system.

the framework are exampled. This includes implementations of state-of-the-art Distributed Constraint Satisfaction, Dististributed Constraint Optimization, Dynamic DCR, Distributed Pseudotree Generation algorithms from the literature. Platforms to be demonstrated include simulation, TCP-based (over an ad-hoc network), and JavaME (communicating over bluetooth). The current problems that are implemented within the system are also demonstrated, including graph coloring, art gallery variants (see Figure 3), the distributed multiple knapsack problem, sensor network problems, meeting scheduling, disaster evacuation planning, and C_TÆMS scheduling. Additionally, a set of metrics are available to automatically compare algorithms. The second component to this demonstration is a brief tour of the development cycle, exemplifying the ease with which an algorithm designer can experiment with, create, and ultimately deploy new algorithms.

3. CONCLUSION

This demonstration presents DCOPolis: a new framework for distributed constraint reasoning researchers to, with relatively little effort, implement and test new algorithms, describe and solve new DCR problems, and simulate and deploy DCR systems. DCOPolis also provides a novel simulation environment that can accurately generate lower-bounds on and asymptotic behavior of distributed runtime. DCOPolis serves to ease the gap of transition from purely simulated agent-based AI to agents that interact within the real world.

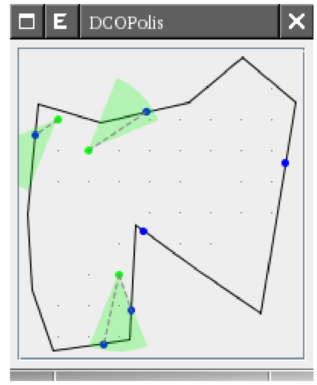

Figure 3: DCOPolis solving a distributed variant of the art gallery problem on a live ad-hoc network.

4. REFERENCES

[1] A. Chechetka and K. Sycara. No-commitment branch and bound search for distributed constraint optimization. In *Proceedings of AAMAS*, pages 1427–1429, 2006.

[2] J. Davin and P. J. Modi. Impact of problem centralization in distributed constraint optimization algorithms. In *Proceedings of AAMAS*, pages 1057–1063, 2005.

[3] A. Haeberlen, A. Mislove, A. Post, and P. Druschel. Fallacies in evaluating decentralized systems. In *Proceedings of the Fifth International Workshop on Peer-to-Peer Systems*, February 2006.

[4] R. Mailler and V. Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *Proceedings of AAMAS*, pages 438–445, 2004.

[5] A. Meisels, E. Kaplansky, I. Razgon, and R. Zivan. Comparing performance of distributed constraints processing algorithms. In *Proceedings of the Third International Workshop on DCR*, 2002.

[6] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. An asynchronous complete method for distributed constraint optimization. In *Proceedings of AAMAS*, pages 161–168, 2003.

[7] A. Petcu and B. Faltings. A distributed, complete method for multi-agent constraint optimization. In *Proceedings of the Fifth International Workshop on DCR*, September 2004.

[8] E. Sultanik, R. Lass, and W. Regli. DCOPolis: A framework for simulating and deploying distributed constraint optimization algorithms. In *Proceedings of the Ninth International Workshop on DCR*, Providence, RI, September 2007.

[9] R. Zivan and A. Meisels. Message delay and discsp search algorithms. *Annals of Mathematics and Artificial Intelligence*, 46(4):415–439, 2006.