RMASBench: a Benchmarking System for Multi-Agent Coordination in Urban Search and Rescue

(Demonstration)

Fabio Maffioletti
University of Verona
Verona, I-37134, Italy
fabio.maffioletti.uni@gmail.com

Riccardo Reffato
University of Verona
Verona, I-37134, Italy
refax88@gmail.com

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

Alexander Kleiner
Linköping University
58183 Linköping, Sweden
alexander.kleiner@liu.se

Sarvapali Ramchurn
University of Southampton
Southampton, SO17 1BJ, UK
sdr1@soton.ac.uk

Bing Shi
Wuhan University of Tec.
Wuhan, China
bingshi@whut.edu.cn

ABSTRACT

This demonstration paper illustrates RMASBench, a new benchmarking system based on the RoboCup Rescue Agent simulator. The aim of the system is to facilitate benchmarking of coordination approaches in controlled settings for dynamic rescue scenarios. In particular, the key features of the systems are: i) programming interfaces to plug-in coordination algorithms without the need for implementing and tuning low-level agents' behaviors, ii) implementations of state-of-the art coordination approaches: DSA and Max-Sum, iii) a large scale crowd simulator, which exploits GPUs parallel architecture, to simulate the behaviour of thousands of agents in real time.

Categories and Subject Descriptors: I.2.11 Distributed Artificial Intelligence: Multiagent Systems.

Keywords: Agent-based simulations, Benchmarks, Urban Search and Rescue.

Online Material: `http://youtu.be/39y6tkhv5O4`

1. INTRODUCTION

The development of benchmarking platforms for agent-based systems is a fundamental step towards building agent-based applications in the real world. Benchmarking platforms allow us to evaluate the performance of state-of-the-art algorithms and mechanisms in controlled settings and determine the best ones to use under a wide range of realistic conditions. To this end, these platforms need to incorporate simulation environments that pose realistic challenges that mimic those of the real world. This means creating environments where the problems are dynamic, the environment variables may be liable to bias or uncertainty and where, potentially, agents have to address issues of scale. Examples of such platforms include multi-agent coordination competition platforms [1, 7] and the TAC platforms [4].[1]

Crucially, such platforms need to allow for easy implementation of new algorithms without requiring researchers to implement low-level elements irrelevant to their algorithm. To date, very few benchmarking platforms have been developed according to such principles (and adopted by the artificial intelligence community). In particular, we note the dearth of realistic testbeds for agent-based coordination mechamisms such as distributed constraints optimisation (DCOP), task allocation (TA), or coalition formation (CF). Those testbeds that do aim to evaluate such algorithms (e.g., DCOPolis[2], CATS[3]) typically provide inputs that are drawn from fixed distributions or define coordination problems that are usually static or require significant extensions to create dynamic settings and large-scale problems. As a result, there are currently no well defined realistic benchmarks for DCOP, TA, and CF.

Against this background, we develop and evaluate a novel testbed for multi-agent coordination algorithms called RMASBench. Our work builds upon the existing RoboCup Rescue simulation platform (RSP) that simulates an urban search and rescue scenario and has also been used by emergency responders and planners to both train and plan for emergencies [6]. However, the RSP as it is, does not allow for easy implementation of coordination algorithms, often requiring coding low-level elements such as network communication protocols and path-planning algorithms. Moreover, the heavy computations involved in generating realistic simulations (e.g., large crowds moving or fires spreading) require multiple machines and only permit simulations with low numbers of agents (fewer than 200). Hence, we significantly extended the RSP in order to allow for the easy implementation of coordination algorithms and develop a large-scale crowd simulator that allows users to run simulations involving thousands of agents in real-time on a single standard computer.

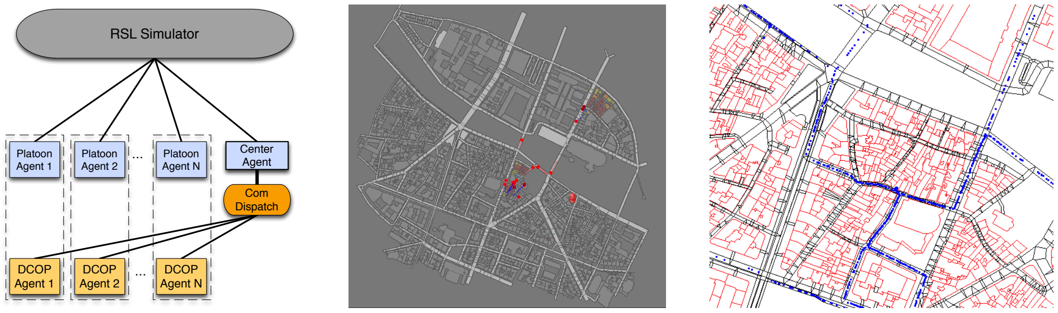
In more detail, our work advances in the following ways the state-of-the-art. First, we provide programming interfaces to implement coordination algorithms without the need of coding and tuning low-level agents' behaviors. Moreover, we provide facilities to compute standard metrics for evaluating coordination approaches. Second, we implemented standard coordination algorithms such as DSA [3] and MaxSum [2] on our test suite, and investigate how they perform under different simulation settings. Third, we develop a large-scale crowd simulator based on the use of Graphics Processing Units (GPUs) that allows us to scale the simulation to thousands of agents.

[1]Altogether these competitions attract hundreds of participants every year and contribute to the development of solutions to real-world problems and the advancement of research in general.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

[2]`http://www.dcopolis.org/`

[3]`http://www.cs.ubc.ca/~kevinlb/CATS/`



(a) (b) (c)

Figure 1: Figure (a), functional connection between the rescue simulator (grey & blue) and RMASBench (orange). Figure (b), a snapshot of the system running. Figure (c), part of the map of Paris with 2000 civilians simulated in real-time on a GPU.

2. THE RMASBENCH SYSTEM

In this section we provide a description of the RMASBench system focusing on the coordination API we provide, the implemented algorithms and the large scale crowd simulator.

2.1 Coordination API and algorithms

RMASBench was developed as part of the RSP to introduce a generic API for multi-agent coordination that essentially provides facilities for exchanging messages among agents and for making coordinated decisions. Moreover, RMASBench provides a library implementing state-of-the art coordination approaches such as DSA and MaxSum, two state of the art DCOP solution techniques. The rational behind this choice is twofold: i) the DCOP framework allows to efficiently solve complex problems by exploiting domain structure and ii) there are a rich wealth of solution techniques for DCOPs (both exact and approximate) that are often compared and evaluated on synthetic benchmarking scenarios (e.g., graph colouring). We believe that in this sense RMASBench is an ideal testing ground to compare such techniques considering various performance metrics (e.g., solution quality, communication and computation overhead) and different operative conditions. Among the several solution techniques for DCOPs we choose these algorithms because they represent two widely used heuristics based on different perspectives: DSA is essentially a greedy local search, while MaxSum is based on inference.

To implement new coordination algorithms in RMASBench, users can simply derive a class from the *DecentralizedAssignment* interface, which essentially represents an agent, and implement the following functions. i) **computeOutgoingMessages()**; ii) **getIncomingMessages()**; iii) **computeAssignment()**. Note that each of these functions will then be called in a synchronized manner within each coordination cycle of the assignment computation. In more details, Figure 1(a) depicts the embedding of RMASBench into the agent simulation package of the RSL while Figure 1(b) shows a snapshot of the system coordinating the operations of fire brigades on a part of the Paris map.

2.2 CUDA-based crowd simulation

At present, the RSP only allows sequential computation for agents' behaviour on a single CPU, hence simulating the movements of a relatively small number of agents can take a consistent amount of time. This imposes severe limitations to the use of RSP both for emergency planners to formulate evacuation plans and for researchers wishing to evaluate their coordination algorithms on large crowds.

To address these issues and equip both practitioners and researchers with an efficient benchmarking platform, we developed an evacuation simulator that implements individual civilians as individual agents running their commands in parallel. Crucially, we implemented agents' path planning and social behaviours in terms of a parallel program that can be run on standard Graphics Processing Units for easy deployment on any PC [5]. By so doing, we are able to exploit the highly parallelised memory and processing architecture of the GPU to speed up the simulations by orders of magnitude and therefore to create realistic situations for decentralised algorithms to solve in real time.
The evacuation simulator was integrated with RMASBench in such a way that both platforms can be run on different PCs to allow for efficient parallel computations at the platform level (espousing the distributed architecture adopted by the RSP). This was achieved by using a socket connection to transmit data between the two. The communication protocol between RMASBench and the crowd simulator is based on XML and allows to exchange information about the simulated agents' positions at every time step. In more detail, RMASBench sends a first message with the initial positions of all the civilians and rescue agents in the map. Figure 1(c) shows a snapshot of an evacuation simulation realised with our CUDA-based crowd simulation with 2000 agents on the Paris map.

3. CONCLUSIONS

We introduced RMASBench, a testbed for multi-agent coordination based on the RSP, which, in contrast to existing testbeds, offers a highly dynamic and large-scale benchmarking environment for Multi-Agent coordination in the USAR domain.

4. REFERENCES

[1] T. Behrens, M. Köster, F. Schlesinger, J. Dix, and J. Hübner. The multi-agent programming contest 2011: A résumé. In L. Dennis, O. Boissier, and R. Bordini, editors, *Programming Multi-Agent Systems*, volume 7217 of *Lecture Notes in Computer Science*, pages 155–172. Springer Berlin / Heidelberg, 2012.

[2] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 639–646, 2008.

[3] S. Fitzpatrick and L. Meetrens. *Distributed Sensor Networks A multiagent perspective*, chapter Distributed Coordination through Anarchic Optimization, pages 257–293. Kluwer Academic, 2003.

[4] W. Ketter and A. Symeonidis. Competitive benchmarking: Lessons learned from the trading agent competition. *AI Magazine*, 33(2):103, 2012.

[5] J. Sanders and E. Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison Wesley, 2010.

[6] N. Schurr and M. Tambe. Using multi-agent teams to improve the training of incident commanders. *Defence Industry Applications of Autonomous Agents and Multi-Agent Systems*, pages 151–166, 2008.

[7] C. Skinner and S. Ramchurn. The robocup rescue simulation platform. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 1647–1648, 2010.