RoboAKUT 2011 Rescue Simulation League Agent Team Description

H. Levent Akın, Yavuz Nuzumlalı, Burak Zeydan, and Mehmet Murat Sevim

Department of Computer Engineering
Boğaziçi University, 34342, Istanbul, Turkey
{akin}@boun.edu.tr
http://robot.cmpe.boun.edu.tr/rescue

Abstract. RoboAKUT is a multi-agent rescue team developed at the Artificial Intelligence Lab of the Computer Engineering Department of Bogazici University. Our primary goal is to build a rescue team based on the market paradigm together with regional task management, then show that this method can be successfully implemented in highly dynamic, multitasking and multi-robot environments. In this paper, we give a detailed description of the software architecture and the algorithms to be used by the RoboAKUT 2011 team. Since in 2010 a new simulator and a new library are released, we have developed new agents for the competition with effective region based exploration, enhanced estimation mechanisms, noise handling, search algorithms and communication layer. The main improvements in RoboAKUT 2011 are the market-driven algorithm implementations and some important tools that help in the development phase such as the program iterator and the standalone fire simulator for market-driven method analysis. Current scores show that RoboAKUT 2011 has an overall 40 percent performance increase compared to RoboAKUT 2010 which to the first place in RoboCup 2010 Agent Competition.

1 Introduction

RoboCup Rescue Simulation agent competition consists of a disaster management simulation with multi-tasking heterogeneous agents (Fire brigades, Fire Station, Police Forces, Police Office, Ambulance Teams and Ambulance Center). In addition to being one of the best test beds for agent coordination, there are many other challenges such as development of agent communication protocols for limited communication and noisy message arrivals, multi-agent path planning, scheduling, optimization, supervised learning for civilian death time and fire behavior estimation and unsupervised learning for agents to develop policies.

RoboAKUT is a multi-agent rescue team developed at the Artificial Intelligence Laboratory of the Department of Computer Engineering of Bogazici University. The team performs rescue operations on the simulation environment provided by the RoboCup Rescue Simulation League. RoboAKUT has been participating in the RoboCup Rescue Simulation Competitions since 2002. Our team has won the **First Place** in the agent competition in RoboCup 2010 Singapore.

The rest of the paper is organized as follows. In Section 2 the team members and their duties are introduced. The contributions made for RoboCup 2011 are given in

Section 3. In Section 4 the Market algorithm and its proposed form of usage for the fire fighters, ambulances and police forces are described. The current scores of RoboAKUT 2011 are given in Section 5. The conclusions are given in Section 6.

2 Team Members and Their Contributions

- Yavuz Nuzumlalı(Developer)
- Burak Zeydan (Developer)
- Mehmet Murat Sevim (Developer)
- H. Levent Akın (Advisor)

3 RoboAKUT 2011

RoboAKUT team code was completely rewritten for the RoboCup 2010 competition. The detail of the improvements introduced are given in [1]. Since the architecture of RoboAKUT 2010 proved to be quite successful, we did not change it for the RoboCup 2011 competition, but instead did some additions to the system. The most important improvements are the following:

- **Task Assignment With Market-Driven Method** is a novel approach that is peculiar to RoboAKUT. In addition to the regional task assignment system, an auction system is being used for the task assignment to each agent. This creates a hybrid market paradigm and regional task assignment system.
- **Testing Mechanism** is a system that is built in order to enable developers in quantitatively comparing results across different implementations with much less effort.

4 The Market-Driven Method

Market-driven methods (MDM) aim the maximization of the overall gain of a group of robots by cooperation, collaboration and/or competition between them. For this aim it is not enough to maximize the profits of all individuals in a group; but, it is necessary to take the total profit of that group into consideration while planning. The key to ”deciding for all” is the communication between robots for trading jobs, power and information. Distributed or centralized decision mechanisms may be used depending on the structures of teams [2].

4.1 Proposed Application of MDM

The proposed improvement on the former system is integration of the MDM and BB methods. This will be achieved as shown in Fig. 1. As can be observed, an extra behavior, compared to the pure behavioral approach, that applies the market logic is inserted to the system. For every task this market implementation will be specialized in order to meet the specific needs of that task.

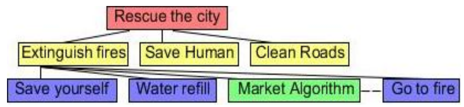

Fig. 1. Market-driven method included into the current behavioral one

4.2 Generalized Market Algorithm and Auction Mechanism

The most important goal in a search and rescue mission is optimizing the process through high level planning. In the RSL competition terms, the optimization measure is the score a team gains. The whole system can be characterized as a "heterogeneous multi-agent system" as there are different agents specialized in different tasks and there is more than one agent contributing to the same aim. The RoboAKUT 2010 team implementation had a behavioral architecture, which makes it good in terms of performance but is far from optimal. Our goal is integrating market driven methods in order to balance reactive characteristics of the system with deliberative ones.

For this purpose, we first started with a general market based algorithm implementation. We determined the most common requirements for a market algorithm. Then we designed a general system containing these required objects in abstract forms. Our abstract system consists of four major components which form the basic requirements of any market algorithm implementation. These are:

- **Visualization tool** provides us with the opportunity of observing the ongoing state of our simulation, i.e., the decisions made, the fires around the city, the placement of agents, and the actions by agents are visualized using this tool. The distinction between the states of objects is achieved through the use of colors and tones of colors. We use this tool for debugging purposes.
- **Auction tool** used for holding auctions, taking bids from the agents, and coming up with results, taking all the data concerning that auction into consideration according to a user-defined function. This general auction mechanism can be modified for the specific needs of different auctioneers.
- **Cost function tool** calculates the costs for alternative tasks given the percepts and past knowledge,. The tool can be considered as a black box that gets inputs and gives out a numeric cost as output. Like the auction tool, this is implemented as a general one that can be modified according to specific requirements concerning different bidders.
- **Communication tool** is used for simulating the imitating communication in the real system for debugging purposes.

4.3 Market Algorithms for Fire Fighters

In a typical fire-extinguishing task, one of the most important missions of an agent is localizing itself around its assigned fire site so as to be able to extinguish it. As a fire disaster around a city proceeds progressively, starting from various points and generally these separate points appear sequentially in the different steps of the simulation, the fire fighter agents tend to group around the starting points of fires. In the market driven approach this does not pose a serious threat but in a simple behavioral implementation where agents hardly are aware of each other, this is a big problem. For, in this case all the agents cluster around the first point of fire and if this is a tough one (expanding and progressing fast) they cannot manage to extinguish many others. This is a problem originating from the ”the grouping problem”. The first fire point, as it is separated from others in time, groups all the agents physically. They all have similar decision tendencies in the rest of the scenario and hence they are inclined to make similar choices. So in other words this clustering tendency can be explained as the physical grouping of agents around fires due to lack of proper communication between them. In MDM the agents do not group as in the left side of Fig. 2 but in a simple behavior based (BB) implementation where the agents hardly aware of each other, this is a major problem; as in the right side of Fig. 2, grouped agents probably miss some other fires.

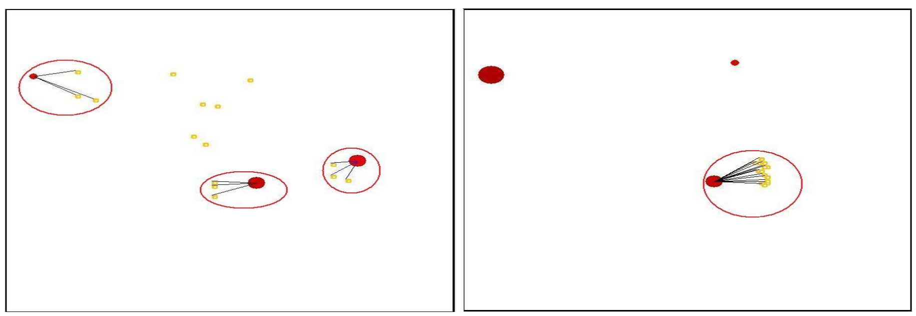

Fig. 2. Screenshots of the test tool (Yellow dots:agents, Red filled circles: fires, Strokes:assignments, Big Red Circles: ”Clustering effect”)

Application of the Market Based Algorithm In the implemented market algorithm, every agent without an assignment calculates the costs for its known fires, and sends the best two of these costs to the center. The center, using its auction tools adds those bids to the appropriate auctions and gathers results for the auctions. If according to the results one agent is assigned to more than one building, an auction weighing the priority of the building and the cost for agent in taking action against that building is held on those results and the final decision is sent to the agent. If according to the results one agent is not assigned to any building, it is added in the auctions held for three buildings with the highest priority and no utilization, and the results involving more than one agent are interpreted using the method described above. During the cycles of central decision, an agent starts its action for the building with the least cost to it and according to the final decision by the center, it either preempts its current action or not. We believe that this algorithm is one of the best alternatives for RoboAKUT as it does not put much strain on the current communication structure and it is easily applicable to the current infrastructure.

As can be seen in the test results in [3], the market algorithm is a very important factor in enhancing the scores establishing the communication hence cooperation and collaboration between agents. It is this collaboration that improves scores, as it avoids ”excessive clustering” around disaster events and provides a close-to -optimum distribution of work, man, and power resources around jobs in an intelligent manner, taking into consideration important factors like collective capacities of a groups versus jobs.

Due to the nature of the search and rescue task there are many parameters that need to be considered, yet as can be seen even from the rather simplistic test cases considered the approach is shown to improve task achievement considerably.

4.4 Application of Generalized Market Algorithm on Ambulance Teams

The market algorithm described in section 4.2 is also used in the allocation of rescue tasks among the ambulance teams. We started using the algorithm with constant cluster size. However we observed that it is not appropriate to use constant clusters for the rescue task. With low cluster sizes, the agents failed to save humans with critical damage, and with high cluster sizes we quickly run out of ambulance teams and humans discovered later are lost if they have a serious damage. In order to solve that problem, we modified the algorithm such that the number of agents in a cluster vary depending on the difficulty of the task. We implemented a model to calculate how many time steps a buried human can live by using the damage and health point of the human. Then, we used that model to determine the number of agents assigned to rescue that human. In the test runs, the agents with the modified algorithm, were able to save about half of the civilians with serious damage. This led to an increase of 1.5 points on the average in the civilian component of the overall score.

4.5 A Market-Based Task Allocation Implementation For Police Forces

Market algorithm implementation was also applied to the police forces. Police forces have a behavior type called ”*ClearPathBehavior*” which defines possible clearing choices as tasks, and determines what to do via comparing the priority of these tasks and choosing one of them.

The previous implementation of this behavior was based on a priority queue system. All tasks have a predefined priority, and the responsibility of *ClearPathBehavior* is to choose the task with the highest priority, and then to make the work needed to accomplish the chosen task. This approach has allowed an easy implementation procedure and also it has provided a relatively good solution to the task allocation problem. But the problem here is that every single agent has its own priority queue and all of the agents choose the local optimal solutions for themselves which rarely results in a global optimal solution as expected. With the help of a market-driven solution, we expect the task allocation will approach to optimal solution without a significant increase in computational complexity. The currently implemented task allocation algorithm for police forces consists of three layers:

- Police Center Layer
- Police Agent Layer
- Communication Layer

Police Center Layer Police Center is the crucial part of the market mechanism. The responsibility of the center is to evaluate the cost values that are determined and sent by the police forces, and to decide an efficient task allocation policy. The police center's work can be divided into three parts, getting cost values from the agents, making task allocation in an optimal manner, and then informing agents about their assigned tasks via the communication system.

The police center has a priority queue for the auction members, and when it receives a cost message, it enqueues the message to the queue according to its cost value, so the head of the queue will be the bidder with the minimum cost value. The current task allocation algorithm is based on allocating the member with the cheapest cost, then looking at the remaining ones and allocating the cheapest one, so on and so forth. After allocating the tasks, the center sends messages to police forces about which task is assigned to which agent, so the allocation is completed.

Police Agent Layer This layer has two parts, sending of the cost values to center, and receiving of the assigned task from the center. The first part mainly includes the calculation of the cost values for each of available *ClearPathTask* according to a predetermined cost function, and sending of these data to Police Center, to participate in auction. For the *ClearPathTask* allocation, auction participation is done as a part of the *ClearPathBehavior*. Currently, a police force only participates in an auction when it is not assigned to a task, or when it has finished its current task. In this case, the agent calculates the related cost values for each of the available tasks, and consecutively sends messages to center which includes info about task, agent, and the cost value.

We embedded the auction participation part into the *ClearPathBehavior* in order to benefit from the efficient decision mechanism of the behavior based architecture of the current system. By this approach, the agents could react to critical events like burning of themselves, so they will maintain their reflexive behaviors.

The cost function for the *ClearPathTask* is currently based only on the distance of the agent to the start location of the related task. Future analyses and development is needed, and will be implemented for the Cost Function in order to make cost determination more realistic and efficient.

The second part consists of receiving and processing of the incoming task assignment message. For each, time step, the agents listen their related channels, and when a task assignment message received, the agent processes the message and controls whether incoming message is for itself, and if so, it takes assigned task and makes it its current clear path task.

Communication Layer We have implemented two new message types to create the necessary infrastructure for the communication between police office and police forces about the task allocation. These are *SendCostsMessage*, and *ClearPathIsAssigned* messages. *SendCostMessage* is the message which provides an interface to police agents to send their relative cost values about the particular tasks. The message contains the data about the related *ClearPathTask*, the *AgentId*, and the *CostValue* for this task.

ClearPathIsAssignedMessage is the message which provides an interface to the police center to inform the police agents about which task it has been assigned. The message contains the data about the assigned *ClearPathTask*, and the *AgentId* of the corresponding agent.

As a future work, we will try to come up with a more realistic cost function which considers different factors like the reason of the clear path requests or whether the requested path is on a main road or a side road. We expect that this cost function calculation will result in a better optimized task allocation between agents.

5 Current Scores of RoboAKUT 2011

Running RoboAKUT 2011 code for all the scenarios in the RoboCup 2010 RSL competition, we obtained important information on the current state of the code. As presented in separate results in Table 1, Table 2, and Table 3 significant improvements in performance over RoboAKUT have been obtained. Considering all the round results together nearly 40 percent improvement is achieved in the overall performance.

Table 1. Preliminary Round Run Results

	Maps									
	Kobe1	VC1	Kobe2	Berlin1	Kobe3	VC2	Paris1	Berlin2	Paris2	VC3
RoboCup 2010	24.692	4.508	52.499	96.05	8.306	24.126	18.075	18.075	16.894	0.721
RoboAKUT2011-r3	40.212	24.666	15.867	52.403	41.686	28.061	16.194	10.033	40.084	0.971
% Change	62.853	447.160	-69.777	-45.442	401.872	16.312	-10.404	-44.493	137.269	34.626

Table 2. Semi-Final Round Run Results

	Maps							
	VC4	Berlin3	Kobe4	Paris3	VC5	Paris4	Berlin41	Paris5
RoboCup 2010	41.133	18.413	100.278	78.603	3.18	2.247	0.951	0.1
RoboAKUT2011-r3	28.247	13.898	82.444	82.685	7.532	1.223	1.110	0.221
% Change	-31.327	-24.520	-17.785	5.193	136.846	-45.573	16.699	121.033

6 Conclusions

In this paper we presented an overview of the agent system model and the algorithms of RoboAKUT 2011 team. They consists of market based agents which can utilize resources effectively even under dynamic conditions for fire fighting, saving civilians and clearing roads. Implementation of a separate simulation system to test variations of market algorithms, and an iterative testing system are the other additional improvements over RoboAKUT 2010.

Table 3. Final Round Run Results

	Maps				
	Paris6	Berlin5	Paris7	Berlin6	Paris8
RoboCup 2010	1.839	16.046	26.144	13.571	45.103
RoboAKUT2011-r3	1.935	7.615	30.114	7.968	17.033
% Change	5.220	-52.545	15.185	-41.288	-62.236

The test runs on the simulator show that the fires are successfully extinguished and the majority of the civilians are saved with an overall 40 percent performance increase over RoboAKUT 2010 which took the first place in RoboCup 2010 competition.

References

1. Orçun Yılmaz and Mehmet Murat Sevim. The development of intelligent agents for robocup rescue simulation league. Technical report, Bogazici University, 2010.
2. H. Kose, K. Kaplan, C. Mericli, U. Tatlidede, and L. Akin. Market-driven multi-agent collaboration in robot soccer domain. In *Cutting Edge Robotics*, pages 407–416. pIV pro literatur Verlag, 2005.
3. Burak Zeydan and H. Levent Akın. Market-driven multi-agent collaboration for extinguishing fires in the robocup rescue simulation domain. In *2nd CSE Student Workshop (CSW'11)*, 2011.