# Improving Rescue Operations in Disasters: Approaches about Task Allocation and re-scheduling

**Silvia Suárez[a], John Collins[b] and Beatriz López[a]**

[a]University of Girona, Av. Lluís Santaló s/n, 17071 Girona, Spain.
[b]University of Minnesota, 4-192 EE/CSci Building 200 Union Street SE, Minneapolis, MN 55455
{sasuarez, blopez}@eia.udg.es; jcollins@cs.umn.edu

**Abstract**

This work concerns allocating tasks to rescue agent teams in a disaster environment in order to mitigate the damage caused by an earthquake. Effective task allocation in this environment is a challenging problem due to several aspects. For instance, its dynamic nature, as in rescue environments tasks arrive in an unpredictable way, and the environment constraints, such as communication flow constraints, as messages can be lost. In addition, agents' capabilities and properties can be diminished due to the damage caused as a consequence of the disaster. In order to tackle task allocation, we use the MAGNET system [1] which provides an interesting allocation framework based on combinatorial auctions. In addition, we have designed and implemented a model for re-scheduling of tasks for agents who are interacting in dynamical and distributed environments. Our experimental framework is the RoboCupRescue simulator [2].

## 1. Introduction

Rescue operations in disaster situations is one of the most serious social issues, and involves very large numbers of heterogeneous rescue teams. In order to provide new technology for giving support in such hostile environments, it is unfeasible to carry out experiments in real-life situations. Therefore, computer simulations offer a valuable platform for testing strategies in advance. One well known simulator is provided by RoboCup-Rescue [2]. Our research is concerned with this simulator. We are specifically interested in allocating tasks to rescue agent teams in a disaster environment in order to mitigate the damage caused by an earthquake.

In Artificial Intelligence (AI) there is a growing interest in using auction mechanisms to solve task and resource allocation problems in cooperative and competitive multi-agent systems. Several systems and frameworks have been developed and applied to different domains and environments in order to deal with this problem. For example, auctions are used in automated negotiation and contracting [3], network bandwidth allocation [4], distributed configuration design, factory scheduling [5], operating system memory allocation [6, 7] and role allocation [8]. Among them we distinguish the MAGNET framework, which tackles industrial procurement, and manufacturing or scheduling areas in a market architecture where the agents are interacting and looking for increase his individual utility as a result of the negotiation of tasks that they are requiring to be performed.

In this paper we test the viability of the MAGNET mechanisms for task allocation in the rescue scenario and present a re-scheduling mechanism required to deal with dynamics characteristic of the environment.

This paper is organized as follows. First, some related work is introduced in section 2. Then, in section 3, we present the description of both MAGNET and the Rescue framework and the communication flow strategy develop for rescue agents. In section 4, the Integration of both systems is explained. In section 5, results are presented and finally, we end with some conclusions and future work.

## 2. Related work.

There are several computational approximations to resource allocation, from the traditional approaches [9] to the ones coming from the more recent advances in the multi-agent paradigm [10]. In the latter, the allocation problem is defined as the distribution of a number of resources amongst a number of agents in order to obtain the maximum benefit either by each agent (competitive scenario) or for the global system (cooperative scenario) [10]. In the rescue domain, the problem concerns task allocation instead of resource allocation, but the same methods can be applied when tasks can be seen as resources associated with a cost rather than a benefit.

Regarding multi-agent approaches, several efforts have been focussed on using economical models to distribute tasks [11]. Then, new terminology has arisen. Particularly, the terms "item", "good" or "offer" are used to name the tasks to be allocated; the terms "buyer" or "supplier" are used as the agents that actively participate in the allocation process, and the term "seller" or "customer" is used as the agent that performs the allocation. Whatever terminology is chosen, the methods developed can be applied to task or resource allocation in many application areas. Particularly, we use the terms tasks (as item, offer or good), rescue agents (as the buyer or supplier), and central agent (as the seller or customer).

For example, regarding the area of industrial procurement, iBundler [12] is a system focused on e-commerce based on agents. It uses combinatorial auctions to choose among several offers in order to optimize the benefit of the buying process. On the other hand, iAuctionMaker [13] is a system which conforms bundles from a set of items. These bundles have to accomplish some properties which are defined based on the knowledge of the experts in a given domain. These properties are characterized as constraints that are used by an optimization algorithm to find the set of bundles that maximize the benefit. Constraints among items are not taking into account in these approaches, while in the rescue domain it is important to take into account precedence relationships. For example, the task of extinguishing a fire is required before the task of rescuing a victim.

One area in which precedence relationships are taken into account is manufacturing in which the term allocation is related to task scheduling. A main issue is the sequentially property characterised in this domain: Each task or process must be done in a given order. One example of this kind of systems is the Fabricare scheduling prototype [5] which presents a multi-agent system for dynamic scheduling of manufacturing orders. Each order represents the set of tasks to be performed, together with the resources they need. Then, tasks agents negotiate with resource agents over deployment of the order, taking into account their agendas, behaviours and due dates. The process followed is centralized and based on demand and time constraints that cause a combinatorial explosion in the number of exchanged messages. The authors, being aware of such communication problem, have developed a protocol that reduces the communication complexity. Even so, in the rescue domain, communication issues are subjected to several constraints and we think that other alternative allocation procedures should be more appropriate.

For example, MAGNET is a framework in which several alternative market allocation procedures can be achieved with three messages. In addition and conversely to other market approaches, MAGNET takes into account precedence relationships among tasks, as required in the rescue domain. Regarding to competitive/collaborative scenarios, MAGNET system is designed to be a multi-agent system of self interested agents who look for his utility. Particularly, MAGNET use combinatorial auctions for task allocation that always finds the best allocation regarding the benefit of the whole system. That is, the total amount of items submitted to the auction appears in the solution. This is an important feature of MAGNET and different from other e-commerce approaches (as for example [14]), and this characteristic is important for the rescue environment due to all the tasks have to be allocated.

Taking into account the previous considerations, and regarding to the rescue scenario we have chosen MAGNET as the framework to deal with the tasks allocation in our domain.

## 3. Systems description.

In this section, basic concepts of the systems involved in our work are provided. Namely, the RoboCupRescue simulator, the communication strategy developed due to the simulator constraints and the MAGNET system.

### 3.1 The RoboCup Rescue system

In any research area it is important to have an experimentation testbed for making tests about the new theories and techniques which are being developed. In this sense, we have been working with the RoboCup Rescue simulator which simulates a disaster environment caused by an earthquake (see Figure 1). The RoboCup Rescue Simulation Project [2] is an international initiative for providing emergency decision support by integrating disaster information, prediction, planning, and the human interface. In this simulated scenario, there are collapsed buildings, fires, blocked highways, people in a state of panic looking for a safe place and rescue agents helping victims. Fire brigade agents, police forces and ambulance teams make up the rescue agents, as well as the central agents: the fire, police and ambulance stations. All the rescue agents have to decide on their actions in order to minimize the disaster damage.

The activities that each agent can carry out are fixed. Ambulance teams are able to rescue civilians (load, rescue, unload commands). Fire Brigades can extinguish fires (extinguish command) and police forces can clear roads (clear command) so that other agents can move.

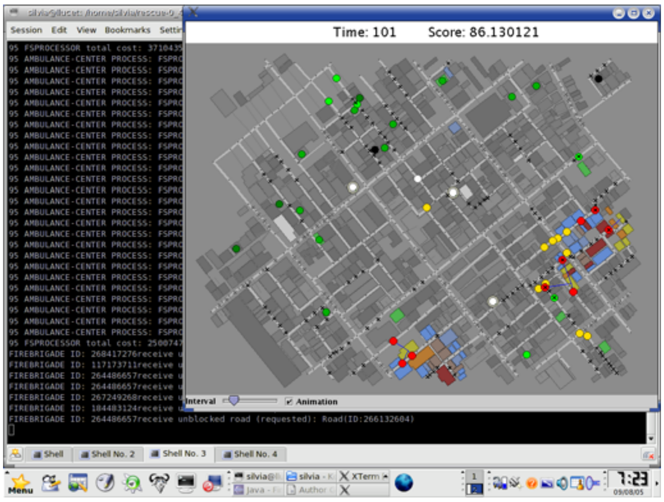



Figure 1. Viewer of RoboCup Rescue Simulator (Kobe Map).

As in a real scenario, the agents have some capabilities constraints. For example, the agents only can see victims located in a 10 meters area (perception constraints). In addition, the agents can send two kinds of messages: either, by voice or by telecommunication, each kind of communication is submitted to several constraints. Regarding voice, agents can hear messages from other agents located in a 30 meters area. Regarding telecommunication , agents can be heard by central agents and rescue agents of the same type as the speaker, it doesn't matter the distance that they are.

One simulation cycle takes one second of computer time, and corresponds to one minute of the real world time. As time passes, the damage increases unless rescue agents actuate. For example, at time n, a fire has a burning degree of k, and at time n+1, the burning degree is k+k' being k'>0, if no fire brigade has extinguished it. In the beginning, rescue agents explore the rescue scenario looking for tasks to carry out (rescue victims, extinguish fires and unblock roads). Once tasks are found, the agents try to solve them and coordination of the agents is necessary. In this sense, the central agents have an important role on the coordination of their rescue team.

Taking into account that coordination is performed by means of direct communication, and that communication is constrained in the rescue scenario, it is necessary to design a communication strategy to facilitate message exchanges. This strategy is explained in the following section.

### 3.2 The RoboCup Rescue communication flow.

Two main kind of communication flow is distinguished: communication for task gathering and communication for task allocation.

#### 3.2.1 Communication for task gathering.

The communication flow presented in figure 2 supports the communication and messages transference between rescue agents and central agents. Every simulation time the central agents are gathering the tasks from the rescue agents. That way, central agents have knowledge about the entire list of tasks inside the scenario, related to their correspondently rescue team.

On the other hand, in the development of their tasks, agents can find new tasks. For example, fire brigades and ambulance team agents find roads that they need be cleared in order to get to the place at which to accomplish their allocated tasks. Then, tasks of police agents are also provided by the fire brigade and ambulance agents by means of the ambulance central and the fire station correspondently (see figure 3).

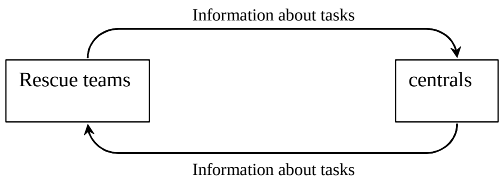



Figure 2. Messages flow about tasks among rescue teams and central agents

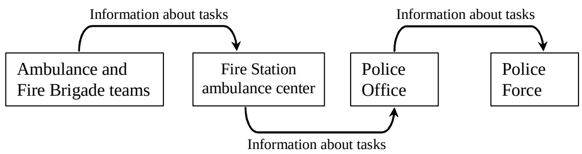



Figure 3. Messages flow about unblocking road tasks from ambulance team and fire brigades to police forces.

#### 3.2.2 Communication for task allocation.

At the beginning of the simulation, all the rescue agents are exploring the scenario looking for tasks. Once agents find a task that agrees with their capabilities, they start to perform it. Otherwise, central agents can assign task to them.

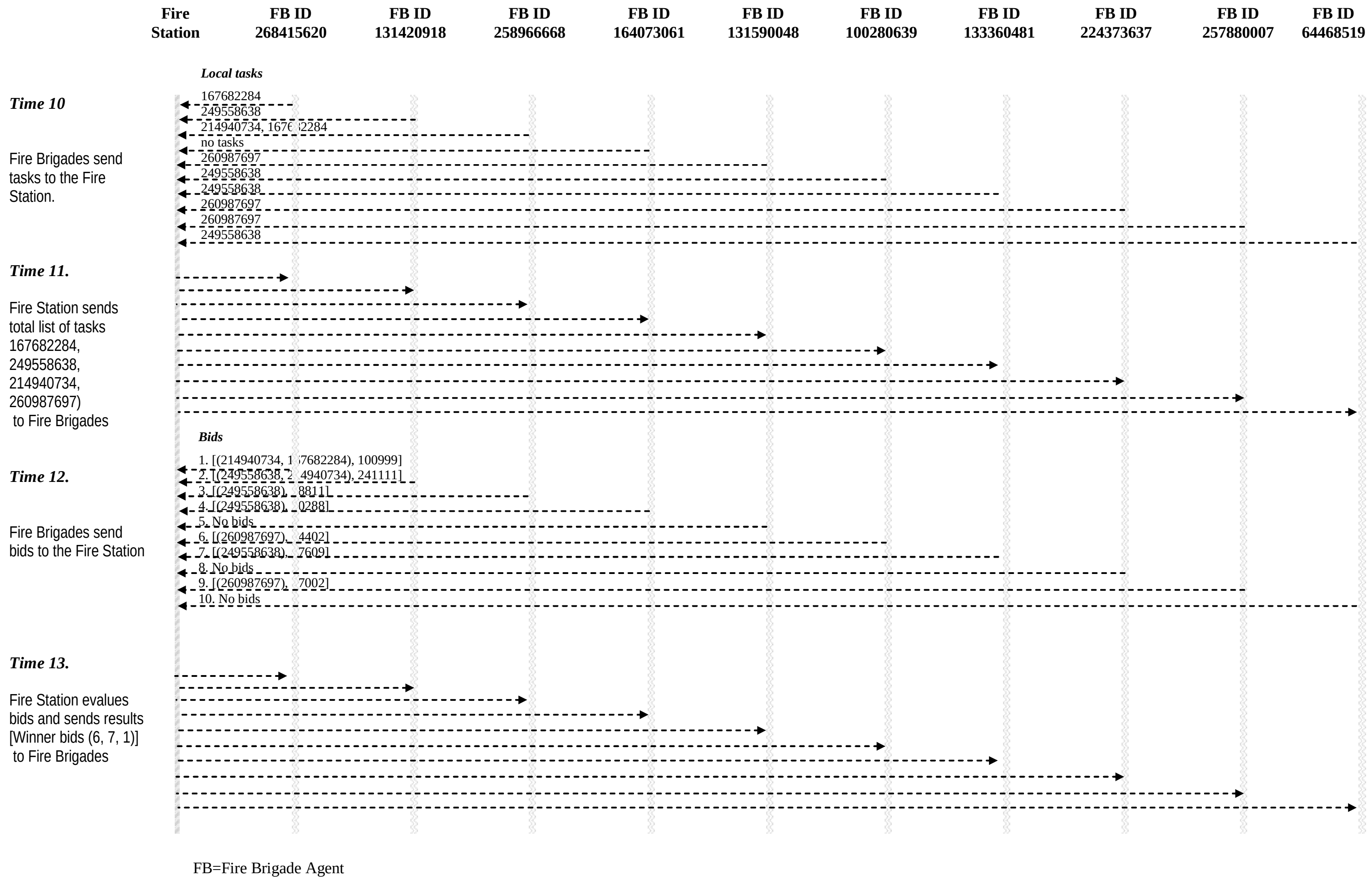



Figure 4. Timeline of the task allocation process for fire brigades and fire station agents.

Central agents are responsible of the task allocation process. In this sense, three parallel allocation processes are being developed, one per each central agents. The ambulance central allocated tasks (rescue victims) to ambulance teams; the fire station allocate tasks (extinguish fires) to fire brigades, and the police office allocate tasks (unblock roads) to police forces. The allocation method chosen is combinatorial auctions [15] (as provide by MAGNET), which requires a communication flow as follows.

1. Agents send the partial list of tasks to central agents (tasks from his perception scope).
2. Central agents send the complete list of tasks to the agents.
3. Agents send bids to the central agents for bundles of tasks. These bids express preferences of the agents by task packages.
4. The central agent selects the winner combination of bids and sends them back to the agents.

According to this communication flow, the task allocation process requires four simulation cycles. In figure 4, an example about a timeline of the process for cycles 10 – 13 and fires extinguishing tasks is presented.

### 3.3 MAGNET System

The MAGNET system is a testbed designed to support multiple agents in negotiating contracts for tasks with temporal and precedence constraints. MAGNET has two main modules, the contracting and the execution module. In the contracting part it is possible to find two kinds of agent roles: the customer and the supplier. The customer needs resources from suppliers in order to carry out their plans. The customer issues Request for Quotations (RFQ) which includes a specification of each task, and a set of precedence relations between tasks. For each task, a time window is specified giving the earliest time that the task can be started and the latest time that the task can be finished. Suppliers submit bids for needed resources in order to do the tasks. Each bid consists of a subset of the tasks specified in the corresponding RFQ, a set of time windows, and an overall cost of the bid. The customer decides which bids to accept by means of combinatorial auction techniques. Each task needs to be covered by at least one bid which is called the Check Coverage constraint. The customer awards the chosen bid combination and specifies the work schedule for the suppliers. Each supplier tries to execute the tasks awarded in the specified time frame.

Next, the process is controlled by the execution module of MAGNET, who takes care of the complexion of the scheduled work. It permits re-scheduling in the case that some provider can not accomplish with his commitments.

Summarizing, for carry out the application of MAGNET framework, the following elements have to be provided:

1. The RFQ's: it specifies a list of tasks and a precedence network among tasks which specifies the order in which the tasks must be executed.
2. The bids from the agents.

## 4. Integrating MAGNET and RoboCup Rescue

The task allocation for rescue scenario is carried out by the central agents. In order to use MAGNET as the mechanism for task allocation, the rescue task allocation problem should be seen as an optimization problem in which the goal is to minimize the damage. Then, the auction mechanism provided by MAGNET can be used, so that central agents are the auctioneers (sellers) and the rescue agents the bidders. The latter submits bids according to the cost of deploying tasks, so the goal of the auctioneer is to find the bid combination that minimizes costs. Then, the information structure required by MAGNET, that is the RFQs and bids, is generated consistently.

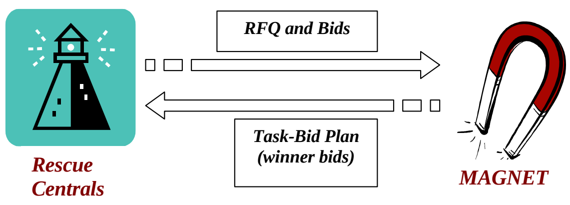



Figure 5. Connection among Rescue-MAGNET systems

The RFQ and Bids are passed from the central agents to MAGNET and his output is a combination of winner bids (see figure 5). The application of MAGNET framework, however, has not been straightforward. We have required to implement a re-scheduling mechanism and to use dummy bids.

### 4.1 RFQ's in Rescue

The RFQ's in the rescue domain are the tasks to be developed which have been gathered by the central agents following the communication strategy presented in section 3.2. For instance, a RFQ would be a list of IDs of burned buildings as follows: [167682284, 249558638, 214940734, 260987697]; we have used dummy time windows for them. In this sense, the dummy time windows express first, that tasks are to be done as soon as possible, and second, that there are no temporal constraints on the start or completion of those tasks.

Regarding, the precedence network for rescue it is presented in figure 6, which is designed for a unique auction process. However, in the RoboCupRescue environment, there is not one central agent with a global view of the entire scenario. Each central agent has to manage his own resources (the rescue agents). Managing just one central agent in charged of the whole allocation leads to increasing entropy in the communication process. Then, three concurrent auctions are required at a time, one per central agent.

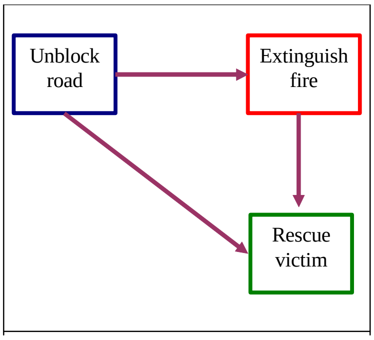



Figure 6. Precedence Network among tasks for Rescue

Tacking into account the reasons exposed above and regarding MAGNET, we are required to send three RFQs, one for each allocation process, and each RFQ only contains part of the tasks network. In this sense, MAGNET doesn't deal with 3 RFQs and 1 tasks network in the same concurrent process. Then, in order to cope with precedence relationships in our domain, a re-scheduling mechanism is required.

## 4.2 Bid Generation

The rescue agents generate bids corresponding to combinations of tasks to be performed in sequential order. Bids are composed by a list of tasks and the cost that the agent has to assume for carry out this list of tasks. Formally,

$$b_i = [L_i, c_i] \quad (1)$$

In order to select tasks for bids, agents take into account that each task should be located in a distance inside a given horizon measure. Only the tasks which are in this horizon are accepted to conform the bids. In our current implementation this horizon measure is 100 meters.

Then, the cost of the bid is related to the distance from the agent $a_i$ to the itinerary that the selected tasks establish. Given, the selected tasks $L_i= [t^i_1,\dots, t^i_n]$; the following distances are computed $d_i= [d^i_1,\dots, d^i_n]$, where $d^i_j$ is the distance between the place of task $t^i_{j-1}$ and $t^i_j$. The first distance $d^i_1$ corresponds to the distance from the agent to the first task $t^i_1$. Finally, the cost is computed as the sum of distances.

$$\sum_{j=1}^{n} d^i_j \quad (2)$$

The cost based on the distance was defined as a first approach. Other factors as size of the burning area and the time fire spread started [16] could be studied in order to calculate the cost of the bid in future work. All bids received by the central agent are processed using the winner determination algorithms of MAGNET. In this sense, the cost of carrying out the rescue tasks is being minimized as result of solving the combinatorial auction [1, 17, 18].

To illustrate with an example the bid generation, let's assume that at time 11, fire brigade ID 268415620 has knowledge about four tasks to develop, For instance, buildings on fire which have to be extinguished. The list of IDs of these buildings is as follows: [167682284, 249558638, 214940734, 260987697].

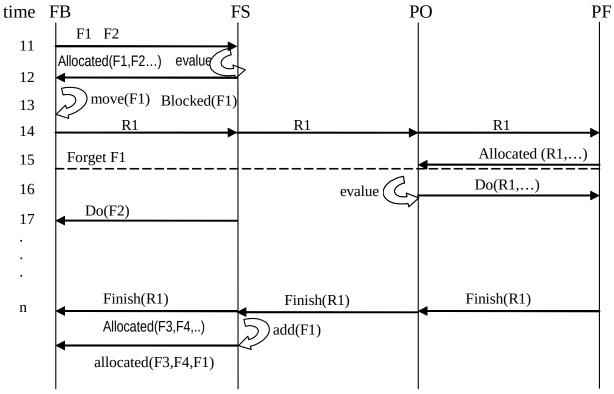
Then, the fire brigade calculates the distance from him self to each building and selects only these that are not more than 100 meters from his location. So, the selected tasks are: building ID 214940734 with a distance of 40.566 from the agent and building ID 167682284 with a distance of 83.890 meters from the agent.

Now, we calculate the cost of the bid as distance from the agent to task 214940734 plus distance from task 214940734 to task 167682284 which is: 40.566 + 60.433 = 100.999. Then, the bid submitted by the agent is the following:

$\text{Bid}_{\text{Agent ID } 268415620}$ = [(214940734, 167682284), 100.999].

## 4.3. Re-scheduling of tasks.

Once tasks are allocated to agents, they try to perform them. There exist several factors that make the agents be unable to execute their tasks, for example, obstructed roads. At this moment a re-scheduling method is needed.




FB=Fire Brigada, FS=Fire Station, PO=Police Office and PF=Police Force

Figure 7. Mechanism for re-scheduling of tasks

For example, suppose that the agent FB have the F1 task in his scheduling of allocated tasks (see figure 7). However, when the agent tries to get until F1, it finds the R1 Road blocked. In this case the agent tries to look for another way to get until his goal. When it is not possible, the agent finds him-self blocked. In order to re-scheduling his tasks, the agent temporally “forget” F1 keeping it in a list of delayed tasks. Next, it sends R1 to the police force as a task to be developed. Then, FB continues with the development of the next task in his schedule. In successive cycles, once R1 is cleared, the agent is informed about that, by the central agent. Then, it re-schedules F1, introducing F1 in his list of pending tasks. Previously, the agent verifies that F1 is realizable, for example, if F1 has grown so much that currently is in-extinguishable, it is not included in the tasks list again and it is forgotten for ever.

The re-scheduling algorithm developed is presented in figure 8. Agents are noted as Ai. Each agent has a list of tasks, Ta= (T1, T2, … Tn) pending to be performed. If the list is empty, the agent explores the environment looking for new tasks. In addition, each agent have a list PTL= (PT1-Td1, PT2-Td2, … PTn-Tdn). This list maps precedence among pair of tasks PTi – Tdi, where PT is the task which precedes the delayed Td task. For example, PT represents the Road’s Id to be cleared and Td a Fire’s Id which has been delayed in the process previously explained.

```
While Ta is not empty

    If Ai is notified about PT cleared from central
        Ai look for Td correspondent to  PT in the PTL list
            If  Td is found
                If Td is realizable
                    Add Td again to his list Ta
                End If
            End If.
    End If

    Select first task in Ta (T1)
    Get T1
      If not Get T1:
          If Ai is blocked in a road Ri
              If there's not other way
                    Add pair R1-T1 inside PTL list
                    Forget Ti (Remove T1 from Ta)
                    Send R1 to be cleared
              End If
          End If
      Else
          Do T1
          Remove T1 from Ta.
      End If
    Next Task in Ta.

End while
```

Figure 8. Algorithm for re-scheduling of tasks

### 4.4 Some implementation issues. Dummy bids

In the Rescue system, after the first auction when some agents are busy (because tasks have been allocated to them) the bids sent by free agents couldn’t contain all the tasks. Therefore, the Check Coverage constraint of MAGNET is not fulfilled and the mechanism doesn’t work.

In order to solve this problem, we issue dummy bids [19] one for each task. Each dummy bid must have a much higher price than any “real” bid. Dummy bids could be part of the solution but just if some rescue agent doesn’t send any real bid for some task. In this sense, when dummy bids are awarded, the tasks corresponding to these dummy bids remain unassigned for the next round in the allocation process.

## 5. Results

We have developed some experiments to evaluate the performance of the task allocation and re-scheduling mechanism developed. Our methods have been applied for the three kinds of agents of the RoboCup simulator (ambulance team, police forces and fire brigades agents).

The RoboCup Rescue simulator configuration used is the following:

Number of agents: 25
Number of fire brigade: 10

| Team number | V value | Team name | Technique description | | |
|---|---|---|---|---|---|
| | | | Police Force | Fire Brigade | Ambulance Team |
| 1 | 83 | Kshitij | ------ | ------ | ------ |
| 2 | 79 | Caspian | Priority assignment for blocked roads. Map division on actuation zones to inform about victims | Priority assignment for buildings on fire. (centralized coordination, there is a leader agent) | Priority assignment for civilians. Centralized coordination in a leader agent. |
| 3 | 69 | Impossibles | Reinforcement learning: For setting the priority of roads | Auctions: Seller-> FireBrigade, Bidders-> Buildings on fire. | ------ |
| | | | Off line learning of the world model | | |
| 4 | 62 | Our approach | Combinatorial auctions and re-scheduling of tasks | | |
| 5 | 58 | MRL | Not available | | |

Number of police force: 10
Number of ambulance: 5
Number of civilian: 70
Number of fire brigade center: 1
Number of police force center: 1
Number of ambulance center: 1
Number of Refuge: 7
Initial point of ignition: 4
Map: Kobe map.

The experiments have been done in the RoboCup Rescue simulator kernel 0.44 version with the new versions of the sub-simulators. We performed 10 simulations and we record the results according to the V score defined in the competition; that is the following:

$$V=(P + S/Sint) * sqrt(B/Bint) \qquad (3)$$

*Where:*

*P: number of living agents,*
*Sint: total HP of all agents at start,*
*S: remaing HP of all agents,*
*Bint: total area at start,*
*B: area of houses that are undestroyed,*

*Where HP is the value related to the damage of the agent, for example, when the civilian is in the collapsed house or it suffers from fire this value is set. Then, hp= damage x T + 10000, where damage coefficient are -100 (in house collapsed) and -1000 (fire), T is elapsed time. When the civilian goes to the refuge, its damage is set 0.*

Our scores obtained are: 63.18; 60.25; 59.79; 65.03; 65.78; 62.06, 62.80, 60.15, 65.10, 63.19 respectively for each simulation test. The higher V value for a map, the better rescue operation (maximum V value is 97).

Then, our average is 62.73. As a reference frame the scores of the four first teams in the final of the latest RoboCup Rescue competition (2005) for the Kobe scenario map are: 83, 79, 69 and 58. So, our agents have obtained a good score in this frame.

In table above, the AI techniques used for the four first teams in RoboCupRescue competition 2005 are presented.

The task allocation mechanism presented is a good alternative to coordination of agents for the rescue operation such as the results showed. It places our team in the fourth position without the use of other strategies as path finding and priority learning that other teams do. So, if in the future we add such strategies, we think that we can improve the result of the operation of our agents.

## 6. Discussion and Future Work

The use of new technologies in rescue operations is a key issue in order to minimize the effects of a disaster. One of these testbed for such technologies is the RoboCup Rescue simulator in which our work is concerned. In this simulator, rescue task should be allocated to rescue agents. In this paper we applied tasks allocation of MAGNET [1] to the rescue domain and we designed and implemented a re-scheduling mechanism. On one hand, MAGNET provides support for task allocation in a market architecture using combinatorial auction techniques. In this framework we have generated a bid generation method for allocating rescue task. On the other hand, our re-scheduling mechanism implements re-allocation of tasks for agents who are interacting in a rescue scenario. It deals with the dynamicity and constraints presented by the environment. Such mechanism permits agents to temporally forget tasks which are not reachable and continue them when the conditions of the environment permit it. Experimental results conclude that our mechanism has a good performance as it is showed for the comparison of our score with the scores in the competition of the RoboCup Rescue of 2005. As a future work, we will tackle bid generation. We are thinking on designing new methods and techniques of bid generation for improving the decision process of our agents when they issue bids. Babanov's works [20] which proposes an approach for soliciting desirable bid combinations to cover tasks, could be useful for reach this goal. The proposed approach finds schedules that maximize the agent's expected utility.

Future research will also include the study of pre-emption mechanisms for tasks that have been assigned to damaged or injured agents in the rescue operation.

## 7. Acknowledgements

This work was supported by the Spanish MEC project TIN 2004-06354-CO2-02 and the FPI DPI2001-2198 mobility grant.

## References

[1] MAGNET System web. Intelligent Agents for Electronic Commerce. http://www.cs.umn.edu/magnet/

[2] RoboCup Rescue Simulator. http://www.robocup2005.org/roborescue/

[3] Collins, J. "Solving Combinatorial Auctions with Temporal Constraints in Economic Agents." PhD thesis, University of Minnesota, June 2002.

[4] Feldmann, R., Gairing, M., L¨ucking, T., Monien, B. and Rode, M. Selfish routing in non-cooperative networks: A survey. In Proc. 28th International Symposium on Mathematical Foundations of Computer Science (MFCS-2003), pages 21–45. Springer-Verlag, 2003.

[5] Sousa, P., Ramos, C., and Neves, J. The Fabricare scheduling prototype suite: Agent interaction and knowledge base. Journal of Intelligent Manufacturing, 14(5):441–455, 2003.

[6] Clearwater Scott H., Market-Based Control: A paradigm for Distributed Resource Allocation. World Scientific, 1996.

[7] Wellman Michael P., A market –oriented programming environment and its application to distributed multicommodity flow problems. Journal of Artificial Intelligence Research, 1:1-23, 1993.

[8] Hunsberger, L., & Grosz, B. (2000). A combinatorial auction for collaborative planning. In Proceedings of the Fourth International Conference on Multiagent Systems (ICMAS-2000), pp. 151--158.

[9] Zweben, M., and Fox, M.S., Intelligent Scheduling. Morgan Kauffman, San Francisco, 1994.

[10] Issues in Multi-Agent Resource Allocation. TFG MARA, 2005. http://staff.science.uva.nl/~ulle/MARA/

[11] Kalagnanam, J. and Parkes, D. C., Auctions, Bidding and Exchange Design, "Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era", Kluwer, Edited by Simchi-Levi, D. and Wu, S.D. and Shen M. ", 2004, Chapter 5.

[12] Giovannucci, A., Rodríguez-Aguilar, J.A., Reyes, A., Noria, F., and Cerquides, J., iBundler: An agent-based decision support service for combinatorial negotiations. In Proc. 19th National Conference on Artificial Intelligence (AAAI-2004), pages 1012– 1013. AAAI Press, 2004.

[13] Reyes-Moro, A., and Rodríguez-Aguilar, J.A., iAuctionMaker: A decision support tool for mixed bundling. In Agent-Mediated Electronic Commerce VI. Springer-Verlag, 2005. To appear.

[14] Leyton-Brown, K., "Resource Allocation in Competitive Multiagent Systems". Ph.D. thesis. August 2003.

[15] López, B., Suárez, S., de La Rosa J.Ll., Task allocation in rescue operations using combinatorial auctions In: I. Aguilo, L. Valverde, M.T. Escrig (eds), Artificial Intelligence Research and Development, pages 233-243, IOS Press, Frontiers in Artificial Intelligence and Applications 100, Netherlands 2003. ISBN 1-58603-378-6.

[16] Habibi, J., Fathi, A., Hassanpour, S., Ghodsi M., Sadjadi, B., Vaezi, H., Valipour, M. "Impossibles Team Description". RoboCup Rescue 2005. July 11-19. Osaka, Japon.

[17] Suárez, S., Resource and Task Allocation in Distributed Environments. A Multi-Agent System Approach. Technical Report. M.Sc. Thesis. October 2004.

[18] Andersson, A., Tenhunen, M., and Ygge, F. Integer Programming for combinatorial auction winner determination. In Proc. Of 4th Int'l Conf on Multi-Agent Systems, pages 39-46, July 2000.

[19] Sandholm, T., Algorithm for optimal winner determination in combinatorial auctions. Artificial Intelligence,135: 1–54,2002.

[20] Babanov, A., Collins, J, and Gini, M., "Harnessing the search for rational bid schedules with stochastic search and domain-specific heuristics." In Proc. of the Third Int'l Conf. on Autonomous Agents and Multi-Agent Systems, New York, July 2004, pp 269--276.