

# LTI Agent Rescue: A Partial Global Approach for the RoboCup Rescue Task Allocation

André H. Pereira <sup>1</sup>

Luis G. Nardin <sup>1</sup>

Jaime S. Sichman <sup>1</sup>

**Abstract:** Coordenação é um dos principais problemas em sistemas multiagentes, além de também desempenhar um papel essencial na gestão de desastres. A alocação de tarefas é uma fase importante do problema de coordenação, já que a decomposição de um objetivo em tarefas é a forma mais natural para se organizar o trabalho entre agentes. Nesse artigo, é proposta uma abordagem híbrida de alocação de tarefas para coordenar os agentes na RoboCup Rescue Agent Simulation, que considera a existência de informações locais e globais. Além disso, é apresentada uma metodologia para a análise comparativa entre times de agentes, que é aplicada na comparação dos resultados obtidos pelo time *LTI Agent Rescue*, que utiliza a abordagem proposta, com os resultados obtidos por outros times de agentes que implementam abordagens distintas para solucionar o problema da coordenação.

**Abstract:** Coordination is one of the key issues in multiagent systems and it also plays an essential role in disaster management. Task allocation is an important phase of the coordination problem, since the decomposition of the objective into tasks is the most natural way to organize work among agents. In this paper, we propose a hybrid task allocation approach to coordinate the agents in the RoboCup Rescue Agent Simulation that considers the existence of both local and global information. Moreover, we provide a methodology to compare agent team results, that is used to compare the results of the *LTI Agent Rescue* team, which uses the proposed approach, with the results of other rescue agent teams that implement different approaches to solve the coordination problem.

## 1 Introduction

Disaster may be described as a crisis situation causing wide spread damage which far exceeds the ability to recover. Its management is normally critical because it involves a very large number of heterogeneous entities in a complex and dynamic environment, where the information is incomplete and uncertain, and decisions must be taken in a timely fashion.

---

<sup>1</sup>Laboratório de Técnicas Inteligentes (LTI) – Escola Politécnica (EP) – Universidade de São Paulo (USP)  
Av. Luciano Gualberto, 158 trav. 3 – 05508-970 – São Paulo – SP – Brasil  
{andre.hahn, luis.nardin}@usp.br, jaime.sichman@poli.usp.br

Moreover, according to Ramchurn et al. [11], the emergency responders are faced with a number of significant challenges when managing major disasters such as (i) the number of rescue tasks being usually larger than the available resources, (ii) the need of different skills to complete each task by its deadline, and (iii) the need for quick resources reallocation.

Therefore, aiming to promote research and development of efficient responses to disaster scenarios, the RoboCup Rescue Agent Simulation<sup>2</sup> competition was proposed in 2001 [8, 7]. The competition involves scoring competing multiagent teams, evolving in a simulated city where an earthquake has just happened. These teams are comprised of heterogeneous rescue agents that interact with the environment and with each other in order to perform their tasks such as rescue civilians, extinguish fires and unblock roads.

In such simulated environment, coordination of heterogeneous agents under incomplete and uncertain information plays an essential role in disaster management and assistance provisioning [7]. In the context of multiagent systems (MAS), coordination ensures that agents will not accidentally interfere with each other's subgoals while attempting to achieve a common goal [13]. Therefore, coordination can be defined as the act of handling the interdependencies between the activities of the agents in order to maintain the system's coherence.

According to Nair et al. [10], the coordination problem in the RoboCup Rescue Agent Simulation can be thought of as a task allocation problem, where the tasks become known at different times and can change over time. Therefore, in this situation the problem can be defined as how to assign the tasks to the different rescue agents in order to obtain the best result in disaster management. Also, according to Nair et al. [10], such problem may be handled using two categories of approaches: *centralized* and *distributed*. In the *centralized* approach, the tasks assignment to the agents, and consequently the agents coordination, is performed by a central component, which needs to have global knowledge about the agents' actions and perceptions. On the other hand, in the *distributed* approach, each agent makes its decisions autonomously and based solely on its own local information.

Although those approaches make the assignment of tasks possible, they suffer from several drawbacks. In the *centralized* approach, the central component is a single point of failure, which can become a potential bottleneck as several concurrently active entities compete for its processing resource. Moreover, this approach is completely based on communication, which usually is a scarce resource in a disaster environment. On the other hand, even not needing neither a central component nor high communication bandwidth, the achievement of a successful coordination of tasks among the agents is very difficult in the *decentralized* approach, since the agents have incomplete and uncertain information to make their decisions. Moreover, agents may not easily locate a task to act upon, which makes them lose too much time searching for this task.

---

<sup>2</sup><http://www.robocuprescue.org>.

Therefore, in this paper we propose a hybrid task allocation approach to the coordination of agents in the RoboCup Rescue Agent Simulation competition, inspired by the *Partial Global Planning* approach [5]. The proposed approach intends to exploit the combined advantages of *centralized* and *distributed* approaches considering and handling the existence of both local and global information, and therefore becoming not completely dependent on communication. Such approach assumes the existence of central components named *center* agents, but the information provided by them is used exclusively when requested. Thus, while the rescue agents keep only local information about the environment, *center* agents keep a consolidated view, based on the data received from all the rescue agents. Moreover, rescue agents prefer to act using their local information and only use information from the *center* agents when they are idle. Our main idea is that *center* agents should not directly coordinate the rescue agents, and these latter should take advantage of the global view provided by the former to help to direct their actions when needed.

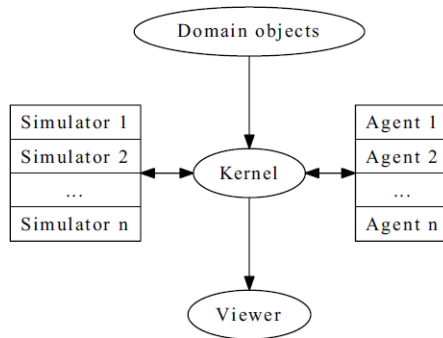
The rest of the document is structured as follows. Section 2 provides an overview of the RoboCup Rescue Agent Simulation competition and the *Partial Global Planning* approach. In Section 3, the proposed *Partial Global* approach for task allocation in the RoboCup Rescue Agent Simulation is described, followed in Section 4 by the description of the *LTI Agent Rescue* team strategy that uses such approach. Section 5 provides a comparative analysis of the participant rescue agent teams of the 2011 RoboCup Rescue Agent Simulation competition, as well as a comparative analysis between the results obtained by the *LTI Agent Rescue* and two other rescue agent teams that implement different approaches. Finally, we present some conclusions and further work in Section 6.

## 2 Background Work

In this section, we provide a brief explanation about the *RoboCup Rescue Agent Simulation* and the *Partial Global Planning* approach.

### 2.1 RoboCup Rescue Agent Simulation

The RoboCup Rescue Agent Simulator is a discrete-time distributed simulation system and it aims to provide a simulation to reproduce conditions that arise after the occurrence of an earthquake in an urban area. In order to make the simulation more realistic, some issues are considered, such as heterogeneity (different types of rescue agents), dynamic environment (fires spread; injured victims), limited information (agents can only see a short distance), uncertain information (agents do not see the true state of the world), limited communication (messages can be dropped or have noise) and limited processing time (agents have a limited time to issue commands). It is thus characterized as a complex multiagent domain [8].



**Figure 1.** RoboCup Rescue Agent Simulation platform architecture [12]

The architecture of the RoboCup Rescue Agent simulator (Fig. 1) is comprised of one kernel and some sub-simulators, each assigned to a specific task within the simulation [12]. There are sub-simulators responsible for fire simulation, traffic simulation, collapse simulation, blockage simulation and one responsible for miscellaneous simulations, such as humans' damage and buriedness and road clearing times. Each sub-simulator runs as an independent process, being the simulator kernel responsible for managing the interaction among them through the network. This setting allows for the computational load to be distributed among several computers, which helps the simulation workload distribution.

The simulator was designed to provide a realistic vision of an urban disaster scenario and its current implementation models a major earthquake in an urban environment where rescue agents are used to control the city's emergency services [12]. In order to better describe its specification, we have subdivided its main features into 3 distinct parts: Environment (Section 2.1.1), Agents (Section 2.1.2) and Communication (Section 2.1.3), described next.

**2.1.1 Environment** The environment represented in the RoboCup Agent Rescue simulator is composed of many different entities. The most important entities in the simulation are buildings, roads, blockages, refuges and humans [9].

*Buildings* represent real constructions and have some properties, such as area, material, fieriness and brokenness. All this information is used by the simulator to calculate the fire propagation and the building's collapse. The building's area and material define how fast the building will burn and how it will collapse. The fieriness property indicates how strong is the fire. The brokenness property indicates how destroyed is the building and how likely it is to crumble.

*Roads* are the entities used by the agents to move along the map. Some of their properties are length, number of lanes and blockages. If a road has no free lanes, i.e. its lanes are blocked by blockages or by other agents parked in the way, it is not possible to move along it.

*Blockages* are entities which block road lanes, hence making impossible their use for the movement of agents. They have a property of repair cost, which defines how hard it is to clear them. Blockages are created by collapsed buildings and can only be cleared by Police Forces (described in Section 2.1.2).

*Refuges* are a special kind of buildings that do not burn or collapse. Moreover, they keep the rescued civilians safe and provide water to the Fire Brigades (described in Section 2.1.2).

*Humans* are humanoid agents, which interact with the environment, whose features and capabilities are described next.

**2.1.2 Agents** The agents in the RoboCup Agent Rescue simulator are divided in two main types: rescue agents and civilians (victims). The rescue agents are still classified into *platoon* agents (moving agents) and *center* agents (fixed agents).

The *platoon* agents are responsible for performing the actions in the environment such as clearing roads, rescuing victims and extinguishing fires. In order to do that, they may perceive situations in their surroundings, which include current situations of buildings, roads and other *platoon* agents that are within their perception radius, and they may exchange messages with other rescue agents through communication channels. Additionally, they may move around the map using the roads, which may be blocked by a blockage or another agent. They are subdivided into three kinds: Ambulance Teams, Fire Brigades and Police Forces.

*Ambulance Teams* are responsible for rescuing civilians. Their exclusive capabilities are to unbury and rescue wounded agents. The Ambulance Teams can rescue any type of agent, not only civilians, although their main goal is to take civilians safely to the refuges.

*Fire Brigades* are responsible for extinguishing fires, therefore their exclusive capability is fire extinguishing. Nevertheless, they have water tanks with limited capacity and there is a maximum distance they can reach with their fire hoses. In order to have their water tanks refilled, Fire Brigades need to go to a refuge and wait for a certain amount of time.

*Police Forces* are responsible for clearing blocked roads. Their exclusive capability is clearing roads. If a road is blocked, agents cannot cross it, so they may get stuck or may need to take a much longer route to get to their destination. In order to clear a blockage, Police Forces need to stay close to the blockage; moreover, the time taken to clear a blockage is variable, depending on its repair cost.

Although each kind of agent performs some exclusive tasks, there is one important difference between them: while it makes no difference to have more than one Police Force to clean a same blocked road, this is not the case for the other two kinds of agents. A fire will be extinguished faster if more Fire Brigades are acting upon it, and victims are going to be rescued faster if more Ambulance Teams are allocated for their rescue. The only limitation of the Ambulance Team is that it cannot carry more than one victim simultaneously.

Table 1 presents the agents capabilities; some are related to direct actions in the environment (move, rescue, load, unload, extinguish, clear), others are related to communication (hear, say, tell) and perception (sense). It may be noticed that civilians do not have any special abilities. Further information about the agents capabilities can be found in [9].

**Table 1.** Agents' capabilities

Type	Capabilities
Civilian	Sense, Hear, Move, Say
Ambulance Team	Sense, Hear, Move, Say, Tell, Rescue, Load, Unload
Fire Brigade	Sense, Hear, Move, Say, Tell, Extinguish
Police Force	Sense, Hear, Move, Say, Tell, Clear
Center	Hear, Say, Tell

On the other hand, *center* agents do not perform any direct action in the environment and they can only interact with other rescue agents through messages. There are three kinds of *center* agents: *Ambulance Center*, *Fire Station* and *Police Office*.

All agents, except the *center* agents, have an initial amount of health points, which may decrease if they are hurt or if they stay inside a burning building. They can also be buried, in which case they are trapped and cannot move until they are unburied.

**2.1.3 Communication** Agents can also exchange messages among themselves through communication channels, which may be affected by communication failures. Therefore, when a message is affected by a failure it may be dropped or received empty in the destination.

The simulator allows the agents to use two types of messages: *tell* and *say*. In order to use the *tell* message type, the agent must be subscribed to a channel and when it sends a *tell* message through this channel, the content is broadcasted via radio to all the *platoon* agents subscribed to it. On the other hand, the *say* message type limits its receivers to all the agents

located within a predefined radius from the sender<sup>3</sup>.

Agents are able to subscribe to a limited number of communication channels whose have limited bandwidth, failure probability and dropout probability.

## 2.2 Partial Global Planning

The *Partial Global Planning* approach was developed by Durfee [5] in order to coordinate the activities of agents in the Distributed Vehicle Monitoring Testbed (DVMT) [3]. In this approach, the cooperating agents exchange information in order to reach common conclusions about the problem to be solved [13]. It is considered *partial* because agents have limited environmental information available in order to make decisions. On the other hand, it is *global* because agents can exchange and obtain relevant information from each other.

In this approach, each agent has a set of *partial global plans* (PGP) that specifies how it will act and interact with others in order to reach cooperation. The agent always uses those plans to coordinate its behavior as best as it can, and its style of cooperation depends on how it forms, exchanges, manipulates, and reacts to PGP. Therefore, coordination emphasizes a sophisticated local control, where the agent uses its local view to control its own actions.

This local control lets agents converge to common plans in a stable environment; however, in dynamic and limited environments, such convergence may not be possible. In such environments, agents continue to respond locally to new situations even when they have inconsistent plans [6]. In order to improve their local control, they exchange their plans among themselves identifying possible interactions among them. More than helping in their local control, such information exchange helps them to improve their integrated coordination.

Thus, coordination entails sharing enough tentative plans that may help one agent to establish a sufficiently global view of its environment in order to recognize how changes to local plans could improve coordination among agents [6].

The *Partial Global Planning* approach may be described as consisting of three iterative stages [13]:

1. Agents locally decide their own goals and generate plans in order to achieve them;
2. Agents exchange information to determine the plans and goals interactions;
3. Agents alter their local plans in order to improve their activities based on plans and goals exchanged through interactions.

---

<sup>3</sup>The default broadcast radius used in the competition is 30 meters.

More specifically, in the *Partial Global Planning* approach the agents may be modeled as a hierarchical structure composed of two levels [4], *individual* and *group*, as described next.

**2.2.1 Individual activity** The agent's *individual* level uses approximate knowledge to quickly characterize very roughly all the possible data interpretations, in order to form goals. Based on those goals, the agent builds tentative *local plans* to attempt to achieve one or more of these goals.

The *local plans* are summarized into a *node-plan* that specifies the goals of the plan, the long-term order of the planned activities, and an estimate of how long each activity will take. This high-level plan represents a long-term strategy that guides detailed planning decisions, and it is the view that is shared with other agents in order to provide an indication of the activities the agent will be engaged in.

A node-plan has also an *activity map*, which is a series of primitive activities, where each activity has a predicted starting time, ending time, and expected result. As the local plan is pursued and changing situations are observed, new details can be added incrementally, but following the same long-term strategy.

**2.2.2 Group Activity** As mentioned, the agents share with other agents their high-level plan's information such as goals, long-term strategy, and rating. When received, such information is processed by the agent's *Group Activity* level which attempts to relate its plan's goals with goals of these exchanged plans. In effect, a partial global planning mechanism uses such goal-relationship knowledge to group plans together and generates mutually constrained and corroborating results, whenever the plans are potentially pursuing some common larger goal. The mechanism then builds a PGP to represent the group goal and the planning activities for achieving this group goal.

### 3 A Partial Global Approach for Task Allocation in the RoboCup Rescue Agent Simulation

In this section, we present our proposed approach, named *Partial Global*, to coordinate the agents in the RoboCup Rescue Agent Simulation by task allocation. Its basic idea was inspired by the *Partial Global Planning* approach described in Section 2.2, mainly because our approach distinguishes incomplete local information (*Partial*) from shared information (*Global*).

However, despite such inspiration in its point of development, the meaning, use, storage and exchange of information are significantly different between both approaches. More specifically, they have the following differences:



- (i) the information handled by the agent in the PGP approach are plans generated or exchanged among the agents, while in our approach they are tasks perceived from the environment or exchanged among agents;
- (ii) in the PGP approach, the exchange of information occurs directly among the agents, while it is performed through a central component in our approach;
- (iii) in the PGP approach, the global plans are stored in the agents, while in our approach the storage is performed by a central component;
- (iv) in the PGP approach, the global information is used to maintain the coherence of local information, while in our approach global information is used to actually decide the agent's goals when it is idle, which means the agent has no local task to act upon.

As already mentioned in Section 1, the proposed approach intends to exploit the combined advantages of the *centralized* and *distributed* approaches considering and handling the existence of both local and global information, thus being independent of communication. Therefore, in order to realize that, the proposed approach enables *platoon* agents to act independently from any *center* agent, acting based solely on their own local information. Such information comprehend those that were sensed in the environment or received from other agents through *say* messages. However, when the agent is idle it is allowed to request information to the *center* agent. Therefore, agents can act according to two different strategies by using these two possible behaviors described above, one strategy completely distributed (*distributed* approach) and another one hybrid (mix both the *distributed* and *centralized* approaches).

The first strategy is used by a *platoon* agents basically when it cannot transmit messages through communication channels (*tell* messages). Thus, its initial behavior is to walk randomly through the environment looking for tasks such as buried victims, burning buildings or blocked roads in order to act upon. When it senses or receives a task that it has the ability to act upon, it does so until the task is solved (e.g. civilian is unburied), the task becomes obsolete (e.g. building completely burnt, civilian is dead), or its resources end (e.g. Fire Brigade water tank is empty). On the other hand, if it senses or receives a task it is incapable to act upon, it stores the information in its memory until it meets a capable agent and transmits such information. In case the agent has more than one task to act upon, it selects the one that provides the best benefit to itself, where the benefit is calculated based on a priority function that uses extra information from the agent and tasks, for instance, distance to the task, burning building area or material, as discussed more in detail in Section 4.

In the second strategy, a *platoon* agent initially acts following basically the same steps of the strategy above. However, since the agent can transmit messages through the communication channels, every time it senses a task, it first transmits the task to the *center* agent

before begin acting upon or store it in its memory for future transmission. The main difference lies in the fact that if the agent is idle, meaning it has no task to act upon, instead of continuing indefinitely to random walk looking for a task to perform, it requests a task to the *center* agent. Since the *center* agent has received the information of all the tasks sensed by all *platoon* agents, it is able to define and transmit back the best task information available to be performed by the requester. Once finished the task execution, the *platoon* agent informs the *center* agent who can then update his global information. Pragmatically, even if the *center* agents cannot directly coordinate the *platoon* agents, the former can perform a sort of coordination when selecting and providing tasks for execution to the latter.

Consequently, we claim that the proposed approach main advantages are its capabilities of:

- Handling situations in which the *platoon* agent has no task to perform, since *platoon* agents can request a task to the *center* agent, if communication is enabled;
- Handling situations in which there is no communication available between the *platoon* and *center* agents, since *platoon* agents are capable to decide and perform tasks by themselves;
- Reducing the required amount of communication, since the agents communicate only when it is strictly necessary.

In the next section, the proposed approach is instantiated in a team of agents for the RoboCup Rescue Agent Simulation.

## 4 LTI Agent Rescue Team

In section, we provide a brief description of the specific behaviors of each type of *platoon* agent (Fire Brigade, Police Force and Ambulance Team) in order to instantiate the *LTI Agent Rescue* team for the RoboCup Rescue Agent Simulation competition.

### 4.1 Fire Brigade Agents

Initially, a Fire Brigade agent walks randomly looking for fire spots in the environment or it may request such information to a Fire Station agent when communication is available.

Since information about a fire spot (target) is obtained, it moves to a location within the fire range where it can start extinguishing the fire. If it is unable to reach the fire location within a predefined amount of time, the current fire is considered unreachable and it is

ignored. Thus, the agent restarts looking for or requests a new fire spot to the *center* agent if it has none in its task list.

On the other hand, if it is able to reach the fire, it starts to extinguish it until one of the following conditions hold: (i) the fire is extinguished, (ii) the building is completely burnt, or (iii) it runs out of water. In the first two situations, the agent stops extinguishing the fire and restarts looking for or requests a new fire spot to the *center* agent if it has none in its task list. Otherwise, when it runs out of water, it finds a path and moves to the nearest refuge to refill its water tank. Once the water tank is refilled, it tries to find a way back to its last target in order to resume its firefighting. This means that the Fire Brigade agent is persistent with its goals, as it does not change its current goal until it is deemed impossible to reach or it is no longer a valid goal.

Furthermore, when the Fire Brigade agent has more than one fire in its local task list, it selects the one that has the highest priority, which is calculated according to the Eq. 1<sup>4</sup>.

$$P = \left( \frac{1}{totalArea} * fieryness \right) + \left( \frac{1}{\sqrt{distance}} \right) + unburningNeighbors \quad (1)$$

Using Eq. 1, it is expected the Fire Brigade to prioritize the building with the smallest area, highest fire intensity, closest to it and with the largest number of neighbor buildings not burning.

The Fire Brigade behavior described above is represented pseudo-algorithmically in Pseudo-Algorithm 1.

## 4.2 Police Force Agents

Clearing the paths to the refuges is a crucial requirement in a disaster environment, since the Fire Brigade agents need to be in one refuge in order to refill their water tanks and the Ambulance Team agents need to have a unblocked path to rescue civilians. Hence, the initial objective of the Police Force agent is to clear a path to the nearest refuge. Therefore, such operation is performed by moving from where the agent stands to the nearest refuge, clearing all the blockages it finds on its way. Once a refuge is reached, the Police Force agent resumes its operation to its general strategy.

Such general strategy consists in walk randomly looking for a blockage or requesting such information to the Police Office agent in case it cannot find any during a certain period of time. As soon as it senses or receives the information about a blockage, the Police Force agent moves to the location and starts to clear it. While clearing a blockage, the Police Force agent transmits a *say* message to warn near Police Force agents about the blockage it is acting

<sup>4</sup>The highest priority maximizes P in Eq. 1.

---

**Pseudo-Algorithm 1** Fire Brigade

---

```
if (agent is on refuge) && (water tank is not full) then  
  Refill  
end if  
if (agent is stuck) || (target is unreachable) then  
  Random walk  
end if  
if (water tank is empty) then  
  Move to refuge  
end if  
if (agent has no target) then  
  if (agent knows fire) then  
    Select target  
  else if (communication is enabled) then  
    Request target  
  else  
    Random walk  
  end if  
end if  
if (target is not in the sight) then  
  Move to target  
end if  
if (target is in the sight) then  
  Extinguish fire  
end if
```

---

upon avoiding goals conflict between multiple Police Forces since there is no advantage of having more than a single Police Force acting upon a blockage.

It may exist a situation in which the Police Force agent has more than one blockage in its local task list, thus it selects the nearest blockage to act upon.

The Police Force behavior described above is represented pseudo-algorithmically in Pseudo-Algorithm 2.

---

**Pseudo-Algorithm 2** Police Force

---

```
if (agent has no target) then  
  if (agent knows blockage) then  
    Select target  
  else if (communication is enabled) then  
    Request target  
  else  
    Random walk  
  end if  
end if  
if (agent is not in the target) then  
  Move to target  
end if  
if (agent is in the target) then  
  Clear blockage  
end if
```

---

### 4.3 Ambulance Team Agents

The Ambulance Team agent randomly walks through all the buildings in the environment looking for victims. As soon as it senses a victim (target), it moves to the target. In the case that exists more than one known victim, the Ambulance Team agent chooses the nearest one to rescue. However, if it is unable to reach the victim within a predefined amount of time, the current victim is considered unreachable and the task is ignored, thus the agent restarts looking for a new victim if it has none in its local task list.

On the other hand, if Ambulance Team agent is able to reach the victim, it starts the rescue procedure. First, it analyzes if the building is on fire; if it is the case, then unfortunately it can do nothing, since it can die in the fire. Otherwise, if the building is not on fire, then it checks if the victim is buried; if it is the case, then it starts the unburying process. After the victim has been unburied, the Ambulance Team agent loads and transports the victim to

the nearest refuge available. Once on the refuge, the victim is unloaded and the Ambulance Team agent restarts looking for a new victim if it has none in its local task list.

The Ambulance Team behavior described above is represented pseudo-algorithmically in Pseudo-Algorithm 3.

---

**Pseudo-Algorithm 3** Ambulance Team

---

```
if (agent is loaded) && (agent is on refuge) then
  Unload victim
else if (agent is loaded) && (agent is not on refuge) then
  Move to refuge
end if
if (agent has no target) then
  if (agent knows victim) then
    Select target
  else
    Walk to next building
  end if
end if
if (agent is not in the target) then
  if (target is buried) then
    Unbury victim
  else
    Load victim
  end if
end if
```

---

## 5 Comparative Analysis

In Section 5.1, we provide first an analysis of the rescue agent teams that participated of the 2011 RoboCup Rescue Agent Simulation League competition, aiming to characterize whether their coordination strategy has an effect on the team's results. Next, we suggest in Section 5.2 a more appropriate statistical methodology for comparing the results generated from a stochastic process, which is the case of the RoboCup Rescue Agent Simulator. Such methodology is then used to perform a comparison among the *LTI Agent Rescue* team performance and other two agent teams: *Sample Agents* team which is provided within the RoboCup Rescue Agent simulator; and *SEU RedSun* team which was the champion of the 2011 RoboCup Rescue Agent Simulation competition.

### 5.1 2011 RoboCup Rescue Agent Simulation Competition Analysis

The 2011 RoboCup Rescue Agent Simulation competition was held at Istanbul, Turkey from 5<sup>th</sup>-11<sup>th</sup> July, and it was comprised of 16 agent teams.

By analyzing the agent teams' Team Description Paper (TDP) [1], we verified that they implement diversified approaches and strategies. Thus, in order to perform a comparative analysis of the coordination approaches used, we classified them as *Centralized*, *Distributed* or *Hybrid* (Centralized & Distributed) approaches and presented them sorted based on the rank result obtained by the team in the RoboCup Rescue Agent Simulation competition (see Table 2). According to the explanation provided in Section 1, in the *centralized* approach the agents actions are coordinated by a central component, while in the *distributed* approach the agent acts solely by using its local information. On the other hand, the *hybrid* approach considers the use of both approaches combined.

**Table 2.** Comparative analysis based on the coordination approach

Rank	Team	Coordination Approach		
		<i>Centralized</i>	<i>Distributed</i>	<i>Hybrid</i>
1	SEU RedSun			X
2	IAMRescue		X	
3	RoboAKUT			X
4	Poseidon			X
5	Ri-One		X	
6	MRL			X
7	SBCe Saviour		X	
8	RMAS ArtSapience			X
9	BonabRescue		X	
10	SUNTORI		X	
11	LTI Agent Rescue			X
12	Brave Circles		X	
13	anct_resq_2011		X	
14	epicenter			X
15	Naito Rescue 2011	Not Available		
16	HfutEngineRescue		X	

Observing Table 2, we can see that none of the teams uses exclusively the *Centralized* approach; this is reasonable, since in situations when communication is unavailable, rescue agents would not work. Therefore, we focus our attention to the *Distributed* and *Hybrid*

approaches, whose are actually evenly used among the teams (8 *Distributed* and 7 *Hybrid*).

Thus, besides the fact that the *Distributed* approach is predominant in the first half of the best agent teams, we could say there is no difference of performance related to the coordination approach used. Such observation suggests that the difference among the agent teams' performance lies more on other approaches and strategies that their agents use to actually act upon a task, such as fire fighting, map exploration and path planning, than on their coordination strategies.

However, we can not confirm such information based exclusively on this former analysis, since during the RoboCup Rescue Agent Simulation competition each agent team is executed only once in each map, whereas the RoboCup Rescue Agent simulator is stochastic. Therefore, we present next the application of a statistical methodology that may be used in order to analyze results from such kind of simulator.

## 5.2 Experimental Analysis

Given the RoboCup Rescue Agent simulator stochastic nature, we think it is most appropriate to perform a more formal and elaborate analysis of the results in order to compare the efficiency of rescue agent teams.

Thus, instead of considering a single run of the rescue agent teams in each map, we made an experiment where we have run each team several times in each map and we have analyzed the obtained results using a statistical method in order to determine whether there is any statistical difference among the rescue agent teams' results.

In order to perform this statistical analysis, we selected three rescue agent teams to compare:

- *SEU RedSun* - This rescue agent team was the champion of 2011 RoboCup Rescue Agent Simulation competition. It uses different approaches to perform the tasks depending on the type of agent. The ambulances adopt a totally centralized decision making approach, while using a particle swarm optimization (PSO) method to predict the time available to rescue a civilian. On the other hand, the Fire Brigades uses a hybrid approach where a particle filter method is used locally to predict the building temperature and inform to its center, which decides how many Fire Brigades are required to extinguish the fire. Meanwhile, a method based on minimum spanning tree is used to coordinate the Police Force agents during the simulation;
- *Sample Agents* - It follows a completely decentralized and greedy approach. It is provided with the RoboCup Rescue Agent simulator and it is quite simple, since the agents just move randomly and perform the tasks as they found them. They do not exchange



any information at all, and they do not implement any kind of optimization in order to select the best task to perform at each moment;

- *LT Agent Rescue* - It uses the *Partial Global* approach proposed in Section 3 and whose agents' strategies are described in Section 4.

Therefore, the main objective of this analysis was to determine if there is any statistical significant difference among the scores obtained from the *SEU RedSun*, the *Sample Agents*, and the *LT Agent Rescue* agent teams.

We have run these experiments using the five maps that were used in the 2011 RoboCup Rescue Agent Simulation competition and were available online<sup>5</sup>, namely Berlin1, Istanbul1, Kobe1, Paris1 and VC1. These maps were chosen since they were particularly planned to explore all the open multiagent research problems (task allocation with uncertainty, coalition formation, cooperation, distributed vs centralized control and communication) [12] and they are available for use by anyone, thus enabling the reproduction of the experiments.

In order to run the experiments, we considered the use of all the three kinds of agents (Fire Brigade, Police Force, and Ambulance Team) as well as all tasks (extinguishing fire, unblocking streets, and rescuing civilians).

In order to enable the comparison between the experiments, the parameters used in all the simulations of the five maps were the same for all the rescue agent teams. All the simulations had the random seed set to 5 and they are all set to perform 300 cycles, where each cycle should take 1 second, where 0.5 second is taken for the agent reasoning and 0.5 second is used for the simulation execution.

There was one voice communication channel and three radio communication channels available. The maximum message size an agent could transmit using the voice communication channel was 256 bytes with a drop out probability of 10% and a broadcast radius of 30 meters. On the other hand, each radio communication channel available had a bandwidth of 1024 bytes with a failure probability of 10% and a drop out probability of 5%. Furthermore, the agents' perception radius was set to 30 meters, the same distance the Fire Brigades could have to extinguish a fire.

We executed each map simulation 10 times for each rescue agent team. Each map simulation was composed of a predefined number of agents specified at Table 3. All the experiments were run in two computers. In the computer with processor Intel i7 (3.00 GHz) and 8 GB RAM, we ran the RoboCup Rescue Agent simulator while in the computer with processor Intel QuadCore (2.00 GHz) and 4 GB RAM, we ran the rescue agent teams.

In order to measure the performance of each rescue agent team, we used the standard

---

<sup>5</sup><http://roborescue.sourceforge.net>.

**Table 3.** Maps configuration parameters

	<b>Berlin1</b>	<b>Istanbul1</b>	<b>Kobe1</b>	<b>Paris1</b>	<b>VC1</b>
Civilian	101	108	150	180	200
Ambulance Team	11	9	20	30	20
Fire Brigade	17	21	15	30	20
Police Force	6	11	20	30	25
Refuge	5	5	2	6	6
Fire Spot	7	0	2	1	9
Ambulance Center	1	1	1	5	3
Fire Station	1	2	1	5	3
Police Office	1	2	1	2	2

RoboCup Rescue Agent Simulation competition score which is computed in Eq. 2, where  $P$  is the quantity of alive civilians at the end of the simulation,  $H$  is a measure of the health condition of alive civilians at the end of the simulation,  $H_{init}$  is a measure of the health condition of the civilians in the beginning of the simulation,  $B$  is the building area left undamaged at the end of the simulation, and  $B_{init}$  is the total initial building area.

$$V = \left( P + \frac{H}{H_{init}} \right) * \sqrt{\frac{B}{B_{init}}} \tag{2}$$

Table 4 presents the mean value and the standard deviation obtained in the execution of the experiments. The full obtained results and the scripts using R Statistics <sup>6</sup> to process the data can be downloaded at [http://www.lti.pcs.usp.br/~gnardin/rital2\\_data.tar.gz](http://www.lti.pcs.usp.br/~gnardin/rital2_data.tar.gz).

The methodology used to analyze the data generated by the experiments is based on the non-parametric statistical hypothesis test named *Wilcoxon's Rank Sum Test* [2]. This hypothesis test was selected because the data did not follow a normal distribution, and hence we could not ignore such data characteristic.

In order to compare the result of the rescue agent teams, it was verified whether the mean value of the scores obtained by using one rescue agent team ( $MV_1$ ) was greater than the mean value obtained by using another ( $MV_2$ ). The underlying idea was that if these results were statistically different, then it would mean that  $MV_2$  is better than the  $MV_1$ . Thus, using

---

<sup>6</sup><http://www.r-project.org>.

**Table 4.** Experimental results

	<b>Sample Agent</b>	<b>LTI Agent Rescue</b>	<b>SEU RedSun</b>
<b>Berlin1</b>	11.94 ± 0.38	11.99 ± 0.30	12.20 ± 0.99
<b>Istanbul1</b>	1.18 ± 0.00	1.31 ± 0.11	1.54 ± 0.00
<b>Kobe1</b>	11.92 ± 0.55	85.49 ± 4.95	102.70 ± 7.48
<b>Paris1</b>	77.64 ± 9.14	78.08 ± 9.21	90.08 ± 12.85
<b>VC1</b>	11.94 ± 0.29	11.58 ± 0.31	17.69 ± 0.54

Wilcoxon’s Rank Sum Test, the following test was performed:

$$H_0 : MV_2 \leq MV_1$$

$$H_1 : MV_2 > MV_1$$

When the hypothesis test was applied to the five maps considering the risk level ( $\alpha$ ) of 0.05, those hypotheses generated the results presented in Table 5. Those results indicate that the  $H_0$  is rejected for four of the maps for *SEU RedSun* compared to *Sample Agent* and for three of the maps for *LTI Agent Rescue* compared to *Sample Agent*.

**Table 5.** Wilcoxon hypotheses test results (p-values)

	<b>SEU RedSun &gt; SampleAgent</b>	<b>LTI Agent Rescue &gt; Sample Agent</b>	<b>SEU RedSun &gt; LTI Agent Rescue</b>
<b>Berlin1</b>	0.485	0.573	0.711
<b>Istanbul1</b>	$7.969e^{-06}$	$1.134e^{-04}$	$3.124e^{-05}$
<b>Kobe1</b>	$5.413e^{-06}$	$5.413e^{-06}$	$5.413e^{-06}$
<b>Paris1</b>	0.004	0.044	0.007
<b>VC1</b>	$5.413e^{-06}$	0.9927	$5.413e^{-06}$

This demonstrates that the proposed approach implemented in the *SEU RedSun* and *LTI Agent Rescue* teams are better than the one implemented in the *Sample Agent* for most of the maps. Moreover, the *LTI Agent Rescue* team performed worse than the *SEU RedSun* team for most of the maps. This result was expected, since the latter was the team that has won the 2011 RoboCup Rescue Agent Simulation competition.

We noticed that for the Berlin1 map, there was no statistically significant differences between the approaches. This happened because the map covers a much larger area, and the number of agents used in the scenario is too low. This fact makes the behavior, and hence the performance, of the *SEU RedSun* and *LTI Agent Rescue* teams approaches very similar to the one obtained by the greedy approach, since the *platoon* agents could neither easily find the civilians nor the burning buildings, and therefore the *center* agents were rarely informed about new data. Consequently, the *platoon* agents practically acted in a decentralized way, using only local information without a global coordination scheme.

Also, we noticed that for the VC1 map, the *LTI Agent Rescue* had no statistically significant differences compared to the *Sample Agent*. In this situation, we observed that the problem was related to the exploration strategy implemented in *LTI Agent Rescue* team which did not take full advantage of the initial provided map information.

## 6 Conclusions and Further Work

In this work, we presented a hybrid task allocation approach to the coordination of agents in the RoboCup Rescue Agent Simulation competition, inspired by the *Partial Global Planning* approach. Such proposed approach considers and handles the existence of both local and global information. While the local information is maintained by the *platoon* agents, the *center* agents maintain a global consolidated view of the environment that was provided by all the formers.

Some experiments were conducted and their results analyzed and presented comparing the *LTI Agent Rescue* team, which implements the proposed approach, and *Sample Agent* and *SEU RedSun* teams. Such analysis indicate that the *LTI Agent Rescue* team performs worse than the *SEU RedSun* team for most of the maps, but better than the *Sample Agent* provided with the RoboCup Rescue Agent simulator for most of the maps. No statistical significance difference was observed in the Berlin1 map among all the teams, since in this map our and *SEU RedSun* teams have practically presented a behavior similar to a pure decentralized team. Moreover, no statistical significance difference was observed in the VC1 map compared to the *Sample Agent*, since our exploratory strategy did not take advantage of the initial provided map information to explore it.

As further work, we intend to improve the strategies of the several kinds of agents, in order to attempt to ascribe to the rescue agents responsibilities for acting in specific parts of the map, depending on the number of fire spots and civilians identified. Also, we intend to improve the agents strategies in order to enable them to make better decisions based on other buildings, roads and victims properties. Moreover, we intend to add into the *platoon* agent an environment global view similar to the one available in the *center* agent in order to allow it to improve its decisions and become more related to the *Partial Global Planning*

approach. By incorporating these improvements, we expect to have a better performance in future RoboCup Rescue Agent Simulation competitions.

## Acknowledgement

André H. Pereira was supported by PIBIC/CNPq/USP. Jaime S. Sichman is partially supported by CNPq and FAPESP, Brazil.

## References

- [1] *Proceedings of the 2011 RoboCup*, 2011.
- [2] S. Boslaugh and D. P. A. Watters. *Statistics in a nutshell*. O'Reilly & Associates, Inc., California, USA, 2008.
- [3] D. D. Corkill and V. R. Lesser. The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. *AI Magazine*, 4(3):15–33, Fall 1983.
- [4] E. Durfee and V. Lesser. Using partial global plans to coordinate distributed problem solvers. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 875–883, 1987.
- [5] E. H. Durfee. *Coordination of Distributed Problem Solvers*. Kluwer Academic Publisher, Dordrecht, The Netherlands, 1988.
- [6] E. H. Durfee and V. R. Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5):1167–1183, sep/oct 1991.
- [7] H. Kitano and S. Tadokoro. Robocup rescue: A grand challenge for multiagent and intelligent systems. *AI Magazine*, 22(1):39–52, 2001.
- [8] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 6, pages 739–743, Tokyo, Japan, October 1999. IEEE.
- [9] T. Morimoto. How to develop a robocuprescue agent for robocuprescue simulation system. Technical report, RoboCupRescue Technical Committee, 2002.

- [10] R. Nair, T. Ito, M. Tambe, and S. Marsella. Task allocation in the rescue simulation domain: A short note. In A. Birk and S. Coradeschi, editors, *RoboCup 2001: Robot Soccer World Cup V*, volume 2377 of *Lecture Notes in Computer Science*, pages 751–754, London, UK, 2002. Springer-Verlag.
- [11] S. D. Ramchurn, A. Farinelli, K. S. Macarthur, M. Polukarov, and N. R. Jennings. Decentralised coordination in robocup rescue. *The Computer Journal*, 53(9):1447–1461, March 2010.
- [12] C. Skinner and S. Ramchurn. The robocup rescue simulation platform. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, pages 1647–1648, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.
- [13] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons Ltd, second edition, 2009.