FUM Team Description: A Modular Architecture for Agent Cooperation

M. Mojtahedi[2], Y. Mohammadi[1], S. Soltanpoor[1],
M.-R. Akbarzadeh-T.[1], M. Khademi[1], M. Kahani[2]
roborescue@yahoogroups.com
[1] *Department of Electrical Engineering,*
[2] *Department of Computer Engineering,*
Ferdowsi University of Mashhad

Abstract. Cooperation and task decomposition is a significant aspect of the RoboRescue problem. Our goal is hence to reach a modular architecture for defining our agents and the mechanism of their cooperation with other agents. In our designation, we consider limitations, restrictions and RoboRescue regulations independent of software implementation. This is to enable easy testing of hybrid intelligent algorithms. In the resulting structure, cooperation among members of one group and cooperation among different groups are considered in both centralized and distributed fashion.

1 Introduction

RoboRescue Simulation [1, 2] provides a dynamic environment with many agents cooperating to do a distributed and difficult task of rescue. Here, we aim to find a modular architecture, which accounts for all restrictions and abilities of agents and the environment, and which is independent from the way it is implemented. The modular architecture allows for implementing and testing hybrid combinations of various intelligence paradigms and mechanisms of learning and adaptability. In particular, neural networks and fuzzy logic are our chosen frameworks of intelligence, and we will use various learning mechanisms such as reinforcement learning (RL) and evolution-based algorithms for learning and optimization. Neural networks are used to compact sensory information, to recognize learned patterns and to learn new patterns. Fuzzy logic, in contrast, is used to allow human-like handling of rules.

The RoboCupRescue problem has several outstanding features and challenges such as distributed task allocation, agent communication, execution time, environmental uncertainties, and losing agents, all of which are characteristic challenges commonly faced in implementation of multiagent systems. The architecture for handling all of these involves distributed decision making to create robustness to local environmental uncertainties as well as centralized decision making to help find globally optimal solutions. In particular, distributed architecture is used here because decision making and sensory information is mostly local, it can be extendable involving fewer or larger number of agents, it is more reliable because of less dependency on communication, and finally, in case of losing few members the system is not devastated. Centralized architecture, on the other hand, provides global decision making to help optimize performance of the overall rescue effort. Centralized architectures usually suffer from a great dependence on a decision making center [3]. The combination of distributed and centralized architecture is aimed to eliminate such weakness in the event of losing a center.

The proposed architecture also involves cooperation among homogenous and heterogeneous agents. The agents act both deliberative and reactive. The advantage of this method is if the communication among agents is disabled, agents could have individual skills to continue on with their rescue effort.

2. Proposed Architecture

The main architecture of the system is shown in Figure 1, as will be discussed below.

2.1 Individual controller

This block is our agents' "brain" and all decisions are made here. The "brain" is responsible for coordinating among different blocks such as: cooperation among members of different groups, cooperation among members of one group, individual decision making, skills, sensors and actuators. Unlike most other architectures which put a high degree of importance to social cooperation, this architecture is more agent-centered and cooperation is rather an emergent behavior of agent decisions/actions.

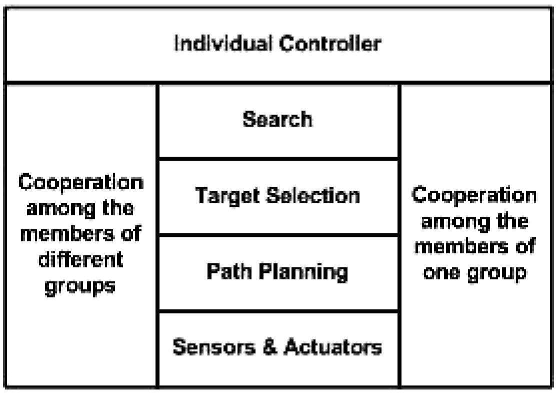

Figure1. Architecture of Agents

2.2 Search

In RobCupRescue Simulation system, location of fires, blocked roads, and injured people are not available globally. Agents are provided with only local information and therefore need to search the environment for recognizing their tasks; as a result, a fast approach is needed to search the environment in a short time. Each agent can search a local neighborhood, do the required tasks in that area, and continue search strategy for new tasks. Because of multiple agents in the environment, it is possible that more than one agent repeatedly search an area. As a result, having a coordinated search strategy is essential. Since communication is limited among agents, intelligence is designed as an emergent behavior of the multiagent system similar to how it is observed in swarm intelligence. Hence, search algorithm is based on Ant Colony. In addition, we aim to use communication among agents for task allocation. The method implemented at this time, however, is based on a random task allocation which does not require communication [4, 5].

2.3 Target Selection

Choosing an appropriate goal can be the most important part. After having researched in one area and finding the tasks, agents make decisions based on parameters that are provided by other modules. One of these parameters is sensory information of agents which sense from the environment. Another effective parameter in this case is the cooperation among different groups. This module shows help requests of other groups, and agents decide whether to answer them or not. The two other inputs of this module come from the module of cooperation among members of one group, one of them being related to distributed cooperation and the other to centralized cooperation. After finishing this module and making the decision and selecting target, we enter the path planning module.

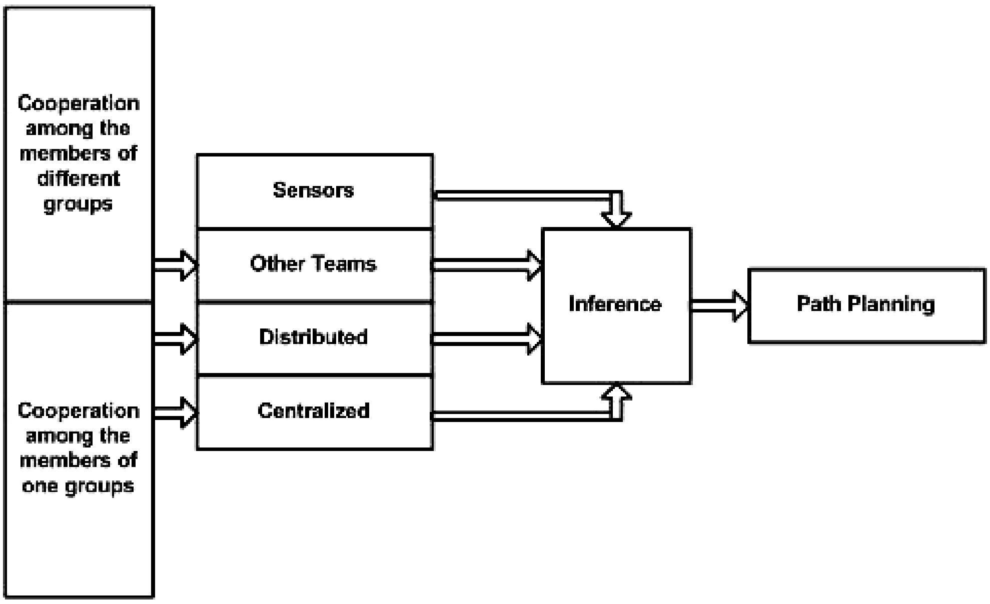

Figure2. Target Selection

2.3.1 ***The target selection in fire agents***

To implement this part we use a hierarchical probabilistic fuzzy system with a GA for optimization of fuzzy parameters.

The sensory information, including fire intensity, building area, type of building, type of houses near fires, their areas and intensity serve as input to 3 Fuzzy systems. After several decision making levels, the final outputs of the fuzzy systems will be the estimated number of agents which are needed to extinguish each fire, indicating its priority.

When an agent can perform more than one task, a roulette fuzzy wheel is used for selecting targets where each target with higher priority and number of required agents, have more chance to be selected. This target is sent to path planning module. Also if this selected target is one of the help requests of other teams, it sends the report to modules of cooperation among different groups. For better performance in fuzzy system, neural networks and learning algorithm is used [6, 7].

2.3.2 ***Target selection in ambulance team***

There is no specific algorithm for target selection of ambulance team. As long as the agents see an injured person, they are expected to heal them.

2.3.3 ***Target Selection in police force***

Police force has the important task of opening the roads and their performance can influence performance of other agent groups by giving them easy access to the roads. Besides the cooperation with other groups, path planning is also another important step for police force. A second layer is responsible for responding to the request of other groups. The algorithm that we wish to implement is explained in more details as follows.

An RL algorithm is used to learn various measures for selecting appropriate target. The output of the RL algorithm provides parts of the inputs to a fuzzy system to select target. Other inputs to the fuzzy system come from the cooperation requests of members of different groups. Furthermore, a GA optimizes the fuzzy system itself for target selection, similar to our fire agents' strategy.

2.4 Path Planning

Finding a suitable path between 2 points in the environment, considering that some of its roads are blocked, is both difficult and important in order to attend to emergencies more adequately. As a result, fast and efficient path planning algorithm is an integral part of a successful strategy. Furthermore, because of interdependencies in rescue mission, cooperation among members of different groups should be considered in design of path planning algorithm. The selected target location and current location provide inputs to this block. Design of the path planning is based on a GP and Fuzzy system [8].

2.5 Cooperation among members of different groups

The cooperation strategy among different groups is defined in this module, and the group centers (fire station, police station, ambulance center) are the basic blocks. Centers that receive messages from other groups' members should eliminate extraneous requests and determine an appropriate strategy to conclude what should occur in a specific situation. Then the center broadcasts the new strategy as request to other members. Each agent, that can fulfill the duty, responds to the centers. (This message shows that the agent would be available for doing the task.)

In this case two possibilities exist. First is if there is no agent to respond. The center should have a queue to refuse or accept requests. The second is if several agents respond to a request. In this case the centers decide which available agents should act and send a message to inform them to do the task.

2.6 Cooperation among the members of one group

This module has two main components, centralized and distributed. We used the centralized cooperation in programming centers, on the other hand used distributed cooperation in cooperation among agents.

Conclusion

In our proposed architecture, our effort was to improve agents' behavior. As a result we used various methods for implementation of different parts of our architecture.
In this case we are working on learning, multiagent learning, swarm intelligence and immunology systems for reaching high efficiency in this environment. [9, 10]

References:

1. The RoboCup Technical Committee: RoboCup -Rescue Simulator Manual Version 0 Revision 4, July 1, 2000.
2. Takeshi Morimoto: How to Develop a RoboCupRescue Agent for RoboCupRescue Simulation System Version 0 1st edition.
3. Poj Tangamchil: A necessity of average reward in cooperation multi robotic learning, Dept. of Electrical and Computer engineering Cornegie Mellon University, Pittsburgh, USA.
4. Eric Bonabeau,Christopher Meyer: Swarm Intelligence: A Whole New Way to Think about business, .May 2001.
5. Kimberly Sharman Evans,Cem Unsal: A Reactive Coordination Scheme for a Many - Robot System, IEEE 1997.
6. Mustafa K.Guven and Kevin M.Passino: Avoiding Exponential Parameter Growth in Fuzzy Systems, IEEE Transactions on Fuzzy Systems Vol. 9. 2001.
7. Edward Tunstel: Mobile Robot Autonomy Via Hierarchical Fuzzy Behavior Control, NASA Center for Autonomous Control Engineering Department of Electrical and Computer Engineering University of New Mexico , Albuquerque, NM 87131 USA.
8. Wolfgang Banzhaf, Peter Nordin, Robert E.Keller: Genetic Programming An Introduction 1997.
9. Surya P.N.Singh, Scott M.Thayer: Immunology Directed Methods for Distributed Robotics: A Novel Immunity - Based Architecture for Robust Control & Coordination, Robotics Institute, Carnegie Mellon University.
10. Jin-Hyung Jun, Dong-Wook Lee: Realization of Cooperative Strategies and Swarm Behavior in Distributed Autonomous Robotic System Using Artificial Immune System, Robotics and intelligent Information System Lab ,Chung Ang Uni., IEEE 1999.