Search-and-rescue using team of robots

Michal Gnatowski

Institute of Fundamental Technological Research
Polish Academy of Sciences
Swietokrzyska 21, 00-049 Warszawa, Poland
{mignat}@ippt.gov.pl

Abstract

In this paper a method of autonomous space searching is described. The task is to search the given area and to find lost objects (e.g. injured people or bombs) as fast as possible. The given area is a building which is divided into rooms. It is assumed that the lost objects are located in groups and finding one object suggests there are more lost objects in the room. Robots divide the area and each robot searches its room. If it finds a lost object it broadcasts the information to whole team and the negotiation mechanism starts. The robots which win the auction, cancel their task and go to the room, where the lost object was found, to find other objects faster. A prototype system was realized in a symulator written in the Java language.

Keywords: Multiagent systems, Robotics, Autonomous systems

1. Introduction

Multiagents systems have wider spectrum of use than single robot systems. There are tasks which can not be done by one robot but more robots can accomplish the task succesfully. A robot is an agent and coordination and cooperation among agents allow to find the best solution of a given task. Tasks given to multirobots systems can be divided into the following groups [2]:

- Grazing - an environment is swept by robots' sensors. For example: vacuuming or search-and-rescue missions;
- Consuming - robots are to find objects, carry them to the proper place;
- Traffic control;
- Moving in a formation - robots are to form a geometric pattern and maintain it;

Multirobots systems path planning can be centralized or decentralized.
In the centralized systems [1][6] there is a central unit which controlls all robots. Increasing robots quantity in the team causes higher cost and lower efficency which is one of the biggest disadventages.
In the decentralized systems each robot plans its path based on local information [3][4][7]. Such systems usually use gradient methods to control the robot [5][8]. The biggest disadvantage of the methods is suffering from local minima and low precision of formation.
Autonomous multiagent systems are also used in deep space exploraton[11]. There are planning agents which due to cooperation and coordination produce partial plans and the partial plans give the planning system which solve given task. In deep space exploration autonomous systems are useful because of communication time delay and limitation of time and cost.
Agents communicate to achieve better goals of

ISSN: 1137-3601.

themselves or of the whole team. A taxonomy of different ways how an agent can coordinate its behaviour is presented at figure 1 [10].

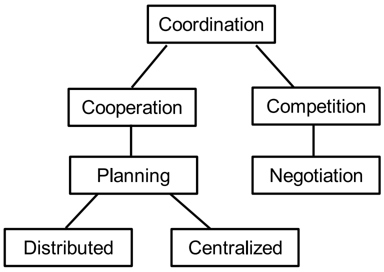

Fig. 1. A taxonomy of different ways of agents behaviour

2. Problem Formulation

In this article a method of auction mechanism which allows to improve efficiency in a group of robots is shown. Let us consider there are objects in a building which may be injured people, bombs, or other objects which are to be found quickly. The task is to find all objects as fast as possible. The assumptions of the task are:

- The team consists of identical robots, without a leader;
- There is no a central unit and the information is completely distributed among robots;
- The team works in a known environment;
- Robots start their work at the same place and finish at any position when the whole building is swept;
- Communication among robots is proved;
- Objects are located in groups and don't change their position during the task;
- Each object has labeled its attendance time which a robot must spend when it finds the object. If lost objects are injured people - attendance time may be a time necessery to deliver first aid to the people;
- Robots have sensors which allow to recognize: (a)lost objects; (b)other robots; (c)any obstacle;
- Robots can be in a *sweeping mode* or *moving mode*. The former means robots go slower and sensors for recognizing lost objest work. The latter means the sensors are off and robots go faster.

A robot which finds a lost object estimates how long will it take to sweep the rest of the room and broadcasts the information to other agents. Each agent which receives information that a lost object was found, calculates the distance to the room with the lost object and compares it with the estimated time of sweeping the room. If it pays - robot cancels its plan and goes to the room with the lost object to help finding other objects faster.
A new term *"step"* was introduced. Going path costs some amount of *"steps"*. Attendance time connected with lost objects is also represented in *"steps"*. This value is better than measuring time, because the system was tested with client-server simulator and features of TCP-IP connections may influence on experiment time. Quality of the system is measured as the average *step* value of finding one object. *Step* value of finding the last object and quantity of won auctions per one robot are also measured.

3. System Architecture

The system has no central unit. Although there is a common knowledge, which is being kept in each agents memory separetly. The details of the communication are described in the paragraph 3.6. To realize the given task robots are to:

1. choose a room;
2. make a reservation of the room;
3. proceed to the room;
4. sweep the room;

The above procedure is performed until either all rooms are checked or an auction is started. If a robot wins an auction it breaks its plan and proceeds to the room where the auction comes from. Before an experiments starts the robot finds the node for every room The method is described below.

3.1. Choosing a room

The rooms have the following labels: *unchecked, reserved, reserved-going-to, being checked, checked*. The label *reserved* means the robot will sweep the room later; *reserved-going-to* means the robot is on the way to the room and is going to sweep it immediately. Other modes are obvious and don't need any comment.

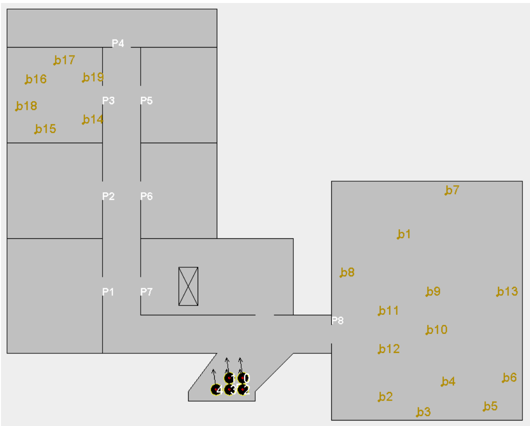

Fig. 2. The team with 5 robots is to check the building. Lost objects are groupped in 2 rooms (P3 and P8)

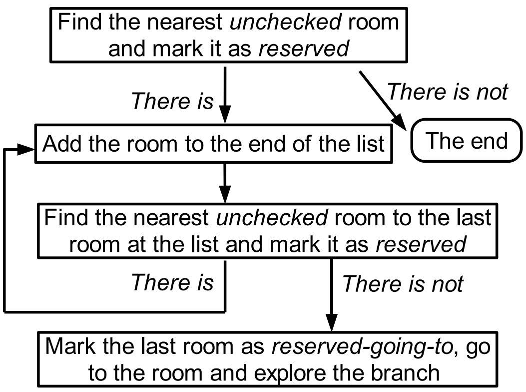

Fig. 3. Making reservation of rooms which are to be checked by one robot

Because of path collision avoidance the team works more efficiently if the robots are dispersed. Thats why each robot tries to reserve its own branch and to explore it from the leave to the root. Robots build a graph with rooms in nodes. The robot which is to choose the room searches the map for the nearest unchecked room. If it finds it, the room is marked as *reserved*, added to the end of the queue and the robot searches the nearest room to the *reserved* room. Such loop is being done until there are unchecked rooms connected the last reserved room.
If the graph has only one branch the first robot reserves all rooms (fig.7, 8). But more popular are buildings where the graph has separated branches (fig.2, 4).

3.2. Going to a room

The path from the current robot position to the room in mode *reserve-going-to* may go through other rooms. To find the path a graph where doors and enlarged walls and obstacles are build for every room. If there is at least one room between the first and the last room than the path consists of subpaths: from the current position to the next room, and from the room to another until the robot reaches the desired room.

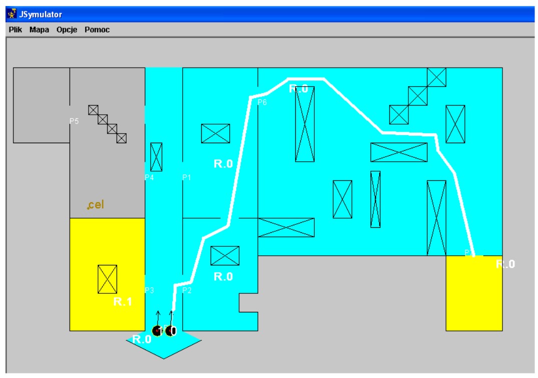

Fig. 4. Path planning to the last room at a branch of a graph. The A* algorithm is used

At the beginning robots build and calculate the distance among doors in every room. To find the path and calculate the cost the algorithm A* is used. Path planning among doors are presented at figure 4, where the path to the last room (yellow one) goes through 3 middle rooms although the path could go through only 2 middle rooms, but the selected path is shorter.

3.3. Sweeping a room

When a robot reaches the door going to the room which is to be checked it starts sweeping the room. The room is represented by rasters, which can be in one of the following modes: *unchecked,* *free, occupied or unavailable.* At the beginning the robot sweeps the room in *ball-like style.* It goes straight as long as there is no obstacle in front of the robot. If an obstacle is detected (wall, another robot or an obstacle inside the room) robots glance off the obstacle and change their direction. Experiments proved that this way of sweeping was efficient if there are a lots of unchecked rasters in the room. Later the diffusion method is used to find the path to the closest unchecked raster. A robot sweeping a room is presented in figure 5.

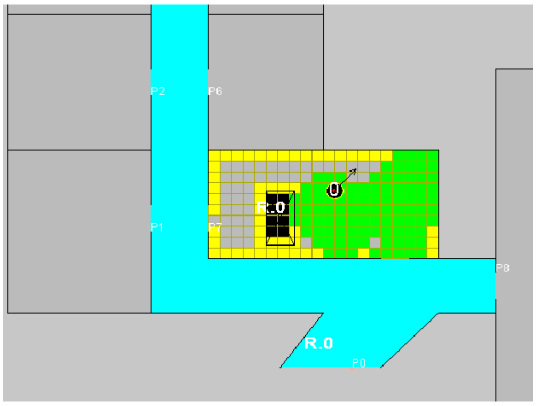

Fig. 5. A robot sweeping a room

3.4. Finding an object

Every object has its attendance cost and is to be attended when it is found. If a robot finds an object it starts the auction and stops further sweeping for as many *steps* as the attendance cost is. This behaviour may simulate calling for help and deliverating first aid to the injured person who was found.

3.5. Auction mechanism

The auction mechanism is the key part of this work. Auctions are used to increase the efficiency of the group. If a robot finds an object it calculates how many *steps* will it take to sweep the rest of the room as shown in equation (1)

$$C_{est} = \frac{S_{un} * C_{ob}}{S_{ch}}, \tag{1}$$

where:
C_{est} - estimated cost of checking rest of the room (given in *steps*),
S_{un} - number of **unchecked** rasters in the room,
S_{ch} - number of **checked** rasters in the room,
C_{ob} - number of *steps* already done in the room;

After that, the robot broadcasts the information to all other robots. The message consists of:

- the ID of the room where the object was found;
- estimate value of how many *steps* will it take to check rest of the room (C_{est});

Robots who receive the above message decide if they want to cancel their current activities and go the the room where the message comes from. They check if the inequotation 2 is satisfied:

$$\frac{C_{est}}{u+1} > C_{app} * \mu, \tag{2}$$

where:
C_{app} - the approach cost - shows the cost of the path from the current position to the room where an object was found;
u - number of robots which are either in the room or on the way to its;
$\mu >= 1$ - cooperation coefficient, shows relation between the estimated cost (C_{est}) and approach value (C_{app}). By default $\mu = 1$. This coefficient is described later in this article;
If the inequality (2) is satisfied the robot is interested the auction and broadcasts its C_{app} value. Otherwise it broadcasts "∞" value. If the robot doesn't answer during fixed *timeout* it means the robot is damaged and it is treated as a "∞" value. The damaged robots are described later in the article.
When all robots send their C_{app} or "∞" value, the robot with the smallest C_{app} wins the auction, cancels its current activities and proceeds to the room where the auction comes from. Other robots increase the value u from the inequality (2) and check if the smallest but one C_{app} satisfies (2). If it is so, the robot also wins the auction. The procedure is being done until the inequality (2) is satisfied. The rest of the robots don't win the auction and continue their activities. The auction mechanism is presented at figure 6.

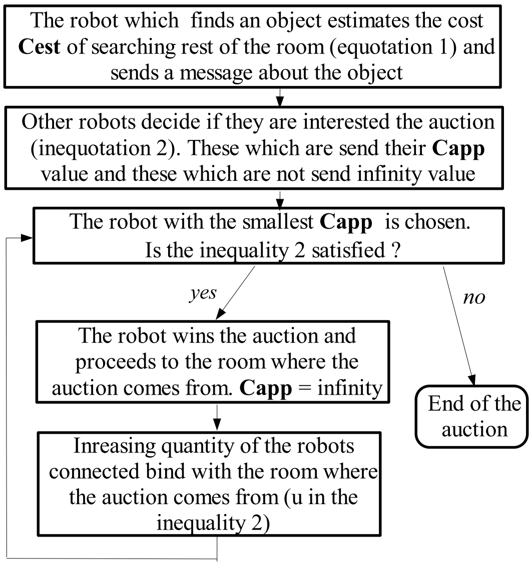

Fig. 6. The auction mechanism

3.6. Communication

The team consists of identical robots without a leader. Robots are synchronized - reserve the rooms and find their paths respectively. This mechanism allows to avoid broadcasting messages by two or more robots at the same time.
Each robot has its number and robots know how many of them are in the team. The communication is realized in the *"token ring"* way, which means the robot which has the *ring* can broadcast any messages (reserving rooms, starting or participating in the auction, etc.) and when it finishes the robot with higher number takes the *ring*. This mechanism allows to avoid a central unit which is one of the biggest advantages of the system!
The individual communication protocol was invented, but it is not described in this article. Communicates which robots broadcast are used to:

- diverse the task;
- collision avoidance;
- sweeping the same room;
- auction mechanism;

Collision avoidance. If a robot notice another robot on its path it may ask the another robot to unoccupie the path. If it is possible, the asked robot moves. A robot may stay at the same place when: (1) ends its task; (2) finds an object; (3) its path is occupied;

Sweeping the same room. Every room is divided into rasters. A raster may be checked or unchecked. When a robot sweeps a room in every *step* it broadcasts messages which rasters were checked. If other robots win auction and go to the same room they know which rasters are unchecked.

Diversing the task and auction machanism were described above detailly.

Robots damage The mechanism of robots damage were not developped deeply, but if a robot doesn't responce when it has the *ring* during the fixed *timeout*, the next robot takes the *ring* and removes the previous robot from the group. The removed robot can't join the group any more. Rooms reserved by the removed robot will be swept at the very end of the experiment. This mechanism is not optimal but allows to accomplish the task.

4. Experiments

To practice the above model a Java simulator was built. The idea of the simulator was based at RoboCup Simulation League [9]. There is a server which simulates environment and clients which simulate robots. Clients send their messages to the server. This messages tell about moving the robot and broadcasting messages to other robots. The server sends to clients information from all sensors (obstacles, other robots and the objects if they are seen) and sends heard messages.

The following experiments have been done:

- 3 different maps (fig.2, fig.7, fig.8);
- the objects were: (a)groupped in 1-3 rooms; (b)evenly placed in all rooms;
- the team consists of 1 - 5 robots;
- different the coefficient μ values from inequality (2). The values μ = 1, 5, 10, 20, 50, ∞ were used;

- each experiment was done 5 times and the average values were taken into further analization;

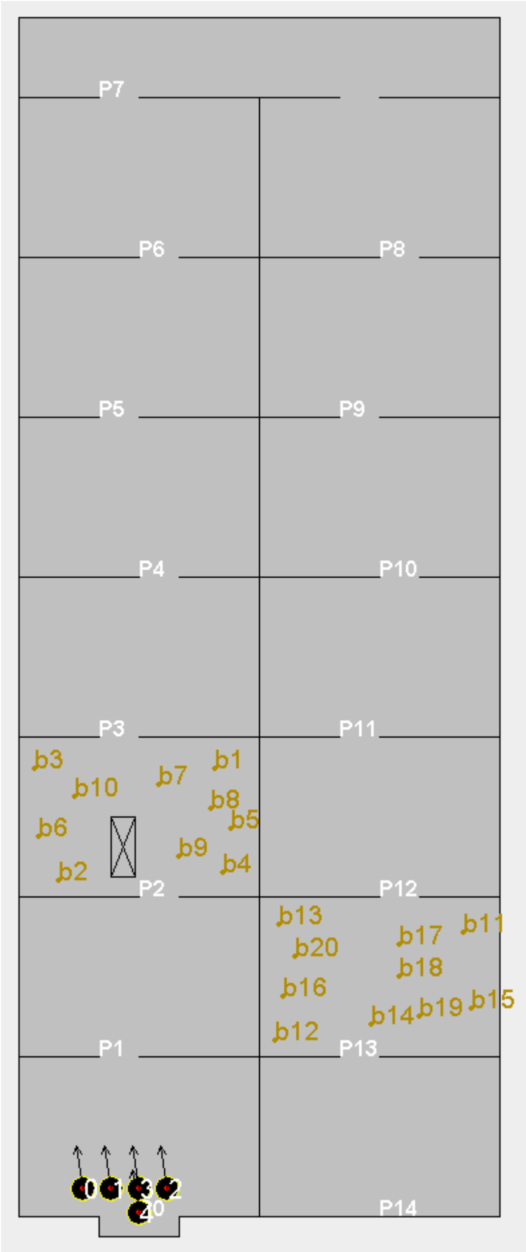

Fig.7. The map which is to be checked (groupped objects)

Robots decide when to cancel their plan and proceed to the room where the auction comes from. Robots check the cost from their position to the room where an object was found and compare it with the estimate cost of sweeping the room where they are, multiplied by μ (inequotation 2). If $\mu = 1$ robots are very sensitive and very willing to change their plans. Such behaviour is not desired. If the coefficient increases the robots behave more assertive and more ofen refuse attending in auctions. On the other hand if $\mu = \infty$ robots do not perform auctions at all what doesn't accelerate finding the objects. The optimize value of the coefficient μ would be very interesting. If the experiment is performed using the same initial conditions the result may be different. It is because room sweeping is randomized and tcp-ip connections may influence to the final result. This is close to reality where we can not assume that doing the same task under theoreticly the same initial conditions gives the same result.

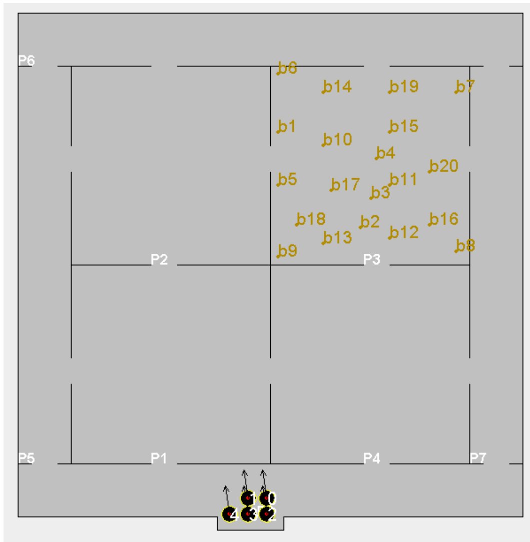

Fig.8. The map which is to be checked. Groupped objects

5. Experimental Results

The average costs of finding objects at the map from fig.2 with groupped and ungroupped objects are presented at fig.9 and fig.10.

If all objects are groupped (here in 2 rooms) decreasing μ value and increasing robots quantity in the team, accelerate accomplishing the task. Figure 11 shows that if auction mechanism is used the objects are found much faster, than without the mechanism.

If all objects are evenly placed in the whole area the auction mechanism doesn't influance or make the results worse (fig.10). When a robot wins an auction and change its plan it means the room where the robot was going to, will be swept later. If objects are groupped there is a high probability the room would be empty. Such behaviour is reasonable. If objects are placed evenly, these placed in rooms which will be swept later, will have to wait longer. In this case more efficient is to switch off the auction mechanism.

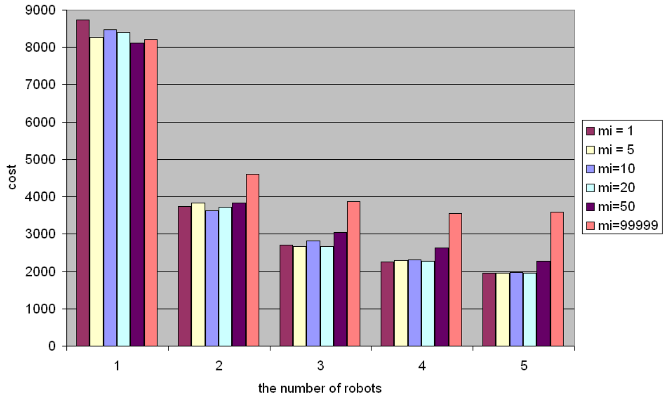

Fig.9. Average value of finding objects from map shown at fig.2. Groupped objects

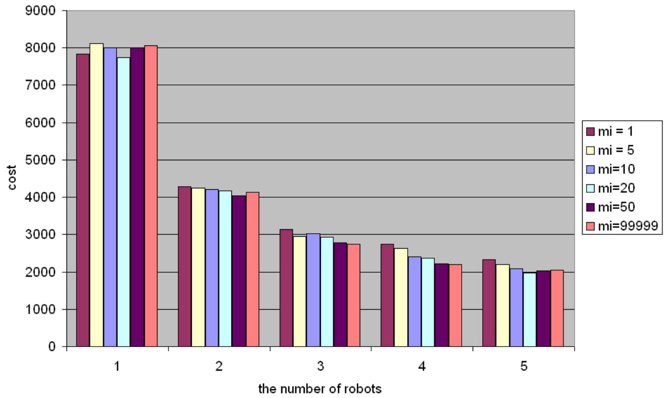

Fig.10. Average value of finding objects from map shown at fig.2. Ungroupped objects

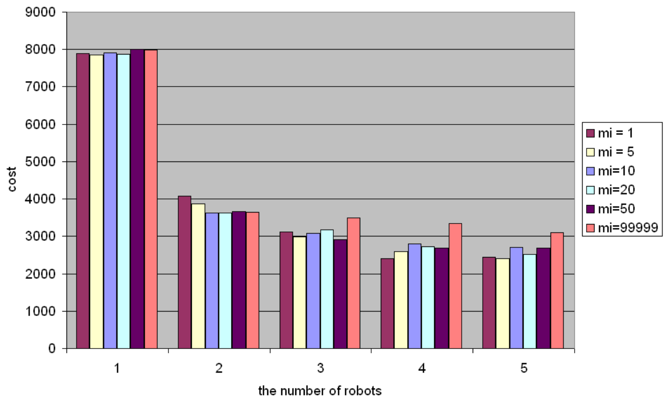

Fig.11. Average value of finding objects from map shown at fig.7. Groupped objects

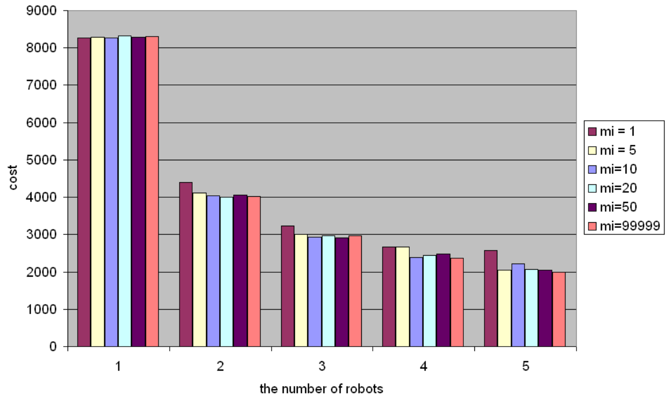

Fig.12. Average value of finding objects from map shown at fig.7. Ungroupped objects

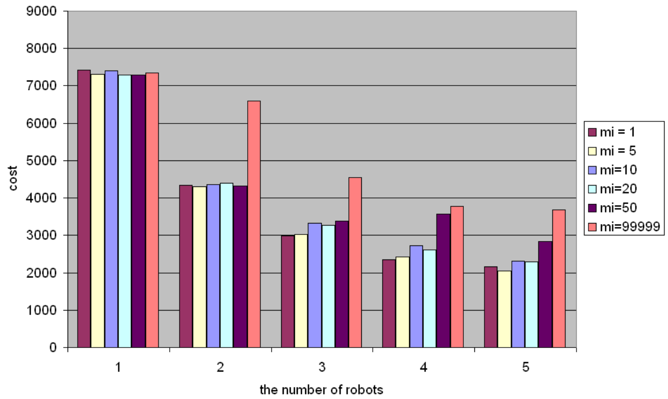

Fig.13. Average value of finding objects from map shown at fig.8. Groupped objects

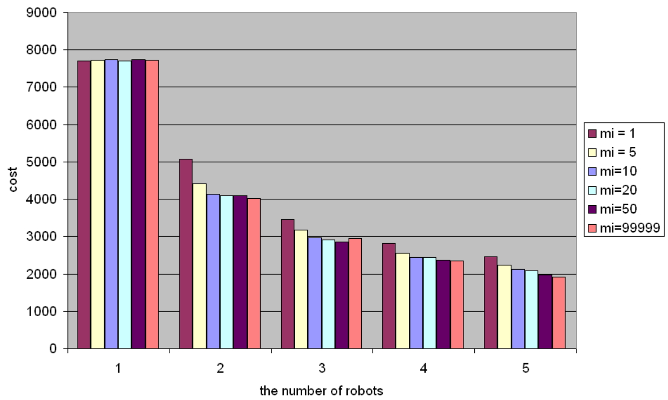

Fig.14. Average value of finding objects from map shown at fig.8. Ungroupped objects

Figures 11-14 show results for 2 other kinds of maps. The results are the same in each case.

6. Conclusion

In this paper a cooperation mechaism of building serching is presented. The task is to find lost objects as fast as possible. The new element in this paper is the auction mechanism used to increase the efficiency of the system. The team has no leader but robots have priorities and their access to the common knowledge database is synchronized. This is a compromise between a group with the leader (when the leader is lost the team can not accomplish its task) and a group of equivalent robots which must negotiate the solution of every subtask (what takes time).
The group is robust. It means if one robot is lost the group is able to accomplish the task.
In the model the *cooperation coefficient* was introduced. The coefficient shows how a robot is willing to cancel its task, change the plan and help others.
This model has been tested at different kinds of buildings with groupped and ungroupped objects and for different values of the *cooperation coefficient*. To test the model 900 experiments have been done.
If lost objects are groupped the auction mechanism increases finding the objects about 25-50 % dependently on the map. For the given map and given quantity of robots it is possible to choose such value of the *cooperation coefficient*, that the task is done faster.

Referencias

[1] Tamio Arai and Jun Ota. Motion planning of multiple mobile robots. *IEEE International Conference on Intelligent Robots and Systems*, 17:261–268, 1992.

[2] R.C. Arkin. *Behavior-Based Robotics.* The MIT Press, Cambridge, 1998.

[3] Kianoush Azarm and Gunther Schmidt. A decentralized approach for the conflict free motion of multiple mobile robots. In *IROS*, 1996.

[4] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[5] B. Barraquand, B. Langlois, and J. C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Robotics and Automation*, 22(2):224–241, 1992.

[6] M. Bennewitz, Burgard W., and S. Thrun. Optimizing schedules for prioritized path planning of multi-robot systems. In *Proc of the IEEE International Conference on Robotics Automation (ICRA)*, 2001.

[7] Yi Guo and Lynne E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *ICRA*, pages 2612–2619, 2002.

[8] Kurt Konolige. A gradient method for realtime robot control. *Proc. of the IEEE/RSJ Interational Conference on Intelligent Robotic Systems (IROS)*, pages 639–646, 2000.

[9] http://www.robocup.org/overview/21.html.

[10] Gerhard Weiss. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence.* The MIT Press, Cambridge, Massachusetts London, England, 2000.

[11] X.Rui, C.Pingyuan, and X.Xiaofei. Realization of multi-agent planning system for autonomous spacecraft. *Advances in Engineering Software*, 36:266–272, 2005.