Cooperation in multirobotics environments

Félix Orlando Martínez Ríos[1] and Carlos Rodríguez Lucatero[2]

[1] Universidad Panamericana campus Ciudad de México
Computer Science Department
fmartin@mx.up.mx
[2] Universidad Iberoamericana
Computer Science Department
carlosr.lucatero@uia.mx

Abstract. In the present article we presented the results of a simulator in order to evaluate the performance of multiagent systems. We approached the problem of exploration of unknown environments using three types of agents: one with ample observation capacities but without moving ability, others with big displacement capacity but whose observation ability is limited to the recognition of their present position (explorers), and finally another group of agents with possibilities of high displacement and load capacity, and narrow sensorial capacity (shippers).
In this work we also present a proposal about paths memorized by agents, based on the creation of a tree of obstacle-free paths. This tree is stored in a blackboard to which all the shipper agents have access, and enables them to choose the best trajectory from their current position to the point in which the samples have been discovered. This work also displays a strategy of collaboration and conflict resolution based on a contract net-like mechanism.

1 Introduction

The problem which we solved with this multiagent system consists on the exploration of an unknown environment [12]. This space is composed by a set of obstacles and samples (objects to be collected) that have to be loaded and bring to a special point which we will call ET^0. We will analyze three different approaches to solve this problem:

1. In the first method we have agents who explore and load samples to the point ET^0 without collaboration.
2. The second approach besides using previous strategy, incorporates collaboration between agents, such that when an agent discovers samples in the environment, when returning to the point ET^0 it leaves landmarks that can be used by itself or other agents to follow this way and then go back to the point where samples were discovered.
3. Our approach is to divide the agents in three different types: the first one with ample calculation and observation abilities (MR^1), the second (MR^2)

© A. Gelbukh, R. Monroy. (Eds.)
Advances in Artificial Intelligence Theory
Research on Computing Science 16, 2005, pp. 89-98

with great possibilities of displacement and capacities of observation limited to its present position and the third (MR^3) with load and displacement possibilities as well as capacities of observation limited to its present position.

Proposals one and two were made by Wooldridge in [12], whereas the third one is our proposal of solution.

1.1 Types of obstacles used in the simulations

In our experiments we used randomly generated obstacles as well as obstacles with some kind of symmetry that makes some subregions of the environment become hard to access by the agents.

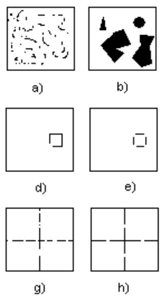

Fig. 1. Obstacles used in the simulations

The obstacles of the Figure 1.a) were randomly generated in all the search space . From now on we will call it type 1 obstacle. We will also identify like this type of obstacle those that are constructed by the user, as it is shown in Figure 1.b), guided by a graphical software. In the obstacle of Figure 1.c) the environment is divided vertically. The agents can move from one side to the other through a small hole placed in the middle of the obstacle. This will be identified as type 2 obstacle. Figure 1.d) presents a small box that completely surrounds the point ET^0. This box has a hole in the left bottom corner through which the agents can leave and go back to ET^0. This obstacle will be identified as type 3. Figure 1.e) displays an obstacle that is similar to the previous one, but in each corner of the box, it has a hole. This will be identified as type 4. The obstacle of type 5, is displayed in Figure 1.f). Like the type 2, has a vertical line that divides the space in two equal sized areas. Unlike the type 2 obstacle this one has several random holes. Figure 1.g) shows type 6 obstacle, made up by two perpendicular lines that divide the search space in four regions of equal dimension. These lines have several random holes that enable the communication between different subregions. Finally, in the Figure 1.h) it can be observed the type 7 obstacle, that like the type 6, has two perpendicular lines but in this case it only has one hole that connect subregions.

2 Experimenting with Wooldridge's proposals

The first solution proposed by Wooldridge [12] consisted in a set of robots that do not communicate to each other and which behavior were basically reactive. Robots leave the main ship (in ET^0 point) and begins to explore using random movements, when an agent finds samples then load them and return to ET^0 following the decreasing gradient field. The other solution given by Wooldridge consisted in a multiagent system with a cooperative behavior (simple but very limited). In that case Wooldridge assumes that the agents return to point ET^0 leaving radioactive landmark in the path. Because we couldn't find numerical results of Wooldridge proposals, we have to simulate his models in order to compare these results with our results.

Previously we did some tests to determine the number of runs necessary to obtain average times that not differ from each other more than 5%.

This strategy of collaboration improves a little the first given solution [2, 6, 7], because it leaves at least a sign of the way to follow from the position of the samples to the ship. Unfortunately if an agent passes over the marks they are erased. In addition there is no guarantee that when arriving at the group of samples at the end of the way they remain there. Another inherent problem is that when arriving at an intersection of ways, there is not a criteria to decide which path must be taken.

From results shown in Table 1 we obtain the following conclusions:

1. For the case of obstacle 1 and 4 both models fulfilled the total of the task in 100% of the simulations, but the time for collaboration case was 23% better for obstacle 1 and 13% better for the obstacle 4, than the time taken by the simulator without collaboration.
2. In the case of type 3 obstacle the time improvement was 28% and in the 100% of the simulation cases task was completed.
3. When using type 6 obstacle the percentage of success in the total fulfillment of the task did not improve remarkably, but the total time was improved in a 28%.
4. For the case obstacle 7 the task fulfillment time was improved as well as the percentage of times that the simulator completes the task until a 100%.
5. Concerning the obstacles of type 2 and 5 their total time was not improved but an increase of 32% and 26% was obtained respectively.

3 Description of our proposal

Our environment consists of two dimension finite space, that will be represented by a matrix ET composed by $n \times m$ cells.

Each element $ET_{i,j}, 1 \leq i \leq n$ and $1 \leq j \leq m$ they represent only one of the following components: empty space, a robot mr_k^l, a number $r \in \mathbb{N}$ of samples, or an obstacle.

The samples located in each grid of the search space, are placed randomly by the simulator.

	without collaboration		with collaboration	
Obstacle type	Time (s)	success percentage	Time (s)	success percentage
Obstacle 1	574	100	445	100
Obstacle 2	1867	54	4205	86
Obstacle 3	1048	96	761	100
Obstacle 4	468	100	411	100
Obstacle 5	1008	87	2006	100
Obstacle 6	994	92	723	93
Obstacle 7	2856	55	2551	81

Table 1. Simulation results of Wooldridge's models with and without collaboration for the seven different obstacle types. We show the total average time to complete the task and the percentage of success in the different simulations

We also have a distinguished element of ET^0 with coordinates i_0 and j_0 that we will defined as starting point and that can be any of the ET cells, with the constrain of not being surrounded by obstacles preventing the access to this point.

We also divided the agents in three classes taking into account: its observation capabilities, processing power [11, 6, 7], displacement abilities and loading capacities [10]. These classes are:

1. Class MR^1: To this class belongs just one agent. It has observation, communication, calculation and storage possibilities, but cannot move. Its observation capabilities enable him to determine if an obstacle-free straight path joining two cells $ET_{i,j}$ and $ET_{u,v}$ exists. Similarly it can store the received information of the agents of class MR^3 (shippers) concerning the obstacle-free straight paths that have been used to reach some $ET_{i,j}$. It also has ample communication capacities that enable him to communicate, as mediator, with all the remaining agents.
2. Class MR^2: Here we will have a set of agents having large displacement and observation capabilities. We will call them explorers. Their processing and storage power are small and its main function is to explore the environment to determine the existence of obstacles and samples. These agents contract the agents of class MR^3 who will make the recollection of the samples. We will call them mr_k^2 agents.
3. Class MR^3: To this last class belong the agents with large loading and displacement capacities but with no observation abilities. These agents will be called shippers and be denoted as mr_k^3. These are in charge to collect the samples and bring them to the ET^0 point. These agents use for their displacement the obstacle-free segments of the path that already have been discovered by other agents of the same class and which are stored in the MR^1 agent of class .

3.1 Behavior of the agents

The explorer agents, in class MR^2, leave the ET^0 point, and move randomly, same as in Wooldrige's model (we focus our attention improving the efficiency of our proposal based in cooperation between agents). If a cell is empty (not occupied by another agent and without obstacles), these agents will move to it. Once in the cell they verify the presence of samples, and if it is the case, then begin the hiring of shipper agents process (belonging to MR^3 class).

Let us suppose that the explorer agent arrive at the cell with coordinates u, v in which it discovers samples, then begins a hiring shipper agents process based on the contract network mechanism [2, 6, 5] and using $KQML$ [9, 3, 8, 4] as the communication language. The agents messages are sent to a blackboard where can be read by the rest of the agents on the system. The agent in MR^1 is charged to support all the blackboard information. Shipper agents that are not currently engaged in a task can read the blackboard to see if they find there hiring messages.

The explorer agents follow a task allocation rule that tries to reduce the number of agents that participate in the recollection. For that reason, once the blackboard was reviewed (agents are ordered decreasingly by their loading capacity) the shipper agents are selected in that order until the amount of samples detected by the explorer agent can totally be loaded. The idea behind this process is to have the smallest possible number of agents moving in the search space and to minimize conflicts produced by crossing paths.

The shipper agents who have been contracted to recollect the samples follow the next sequence of steps:

1. With the information stored in the MR^1 class agent, it determines if between the points of coordinates u, v and i_0, j_0 an obstacle-free straight path exists. If it exists then it follows the straight line segment that join them and publishes in the tree of discovered paths.
2. If it does not find a straight path in the previous step, then it begins to consult the information stored in the tree of discovered paths. Whenever it arrives at a node of this tree it follows the same behavior of the step 1, to try to arrive at u, v. This process continues until it finds a road. In this case a new obstacle-free segment is added to the tree. Otherwise the tree of discovered paths overflows and the task is rejected.
3. If the number of rejected tasks exceeds a given threshold, then the shipper agents move randomly trying to achieve a point where the recollection task can be continued and apply the step 1. If this process also fails then the task is kept in the blackboard for later accomplishment.

3.2 The tree of discovered paths construction

In order to understand how the tree is constructed a hypothetical scene is given as example in Figure 2.

The node labeled by 0 corresponds to ET^0, the points labeled by $1, 2, 3, 4, 5$ and 6, correspond to cells in the neighborhood where there is a certain number

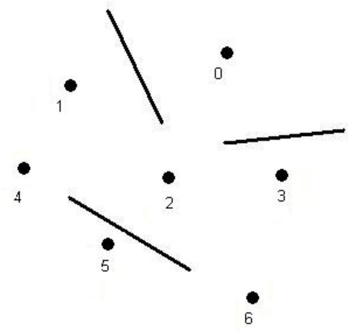

Fig. 2. Example of construction of the tree of obstacle-free straight paths

of samples to gather. Finally, the segments of straight lines represent obstacles. Initially the tree is empty. Let us suppose that an explorer agent arrives at point labeled 2, and at this point we have an obstacle-free straight path. Then shippers will arrive at this point and will store a first node in our tree having the coordinates of the achieved point. At this stage the tree is rooted at the 0 node, the coordinates of the point are stored in the node and a descendant node labeled by 2 is added to it (the coordinates of point 2 are stored in the corresponding node). This construction stage is displayed in Figure 3.a).

Later the explorer agents discover points 1 and 3, in this order. Given that they are not reachable by an obstacle-free straight path from 0, then the information of the tree is consulted and it is observed that point 1 can be reached from point 2. This new path is added to the tree as it is shown in the Figure 3.b). Similarly the path to the node 3, from node 2 is added, as it is shown in the Figure 3.c). After that, points $4, 5$ and 6 are discovered, in that order, and added to the tree as it is shown in the Figure 3.c). It is important to notice that point 5 cannot be reached before discovering point 4 or 6. It's clear that this is not a binary tree because more than two paths can be added to the same node.

Many trees can be constructed for the same environment (depending how samples are discovered). This is not and issue because the tree only is useful to access new locations based on previously known locations and not to describe the environment itself.

This mechanism can fail if there is no reachable point from ET^0 . In order to avoid this problem the simulator is equipped with a positive integer value representing the maximum number of allowed failures. When this value is reached the shipper agents make a first random walk [1]. After that the initial algorithm is retaken.

3.3 Conflict negotiation between agents ready to collect samples

When an explorer agent mr^2_m detects samples, it sends a hiring message to all the shipper agents. This message is attended by all the shipper agents which are

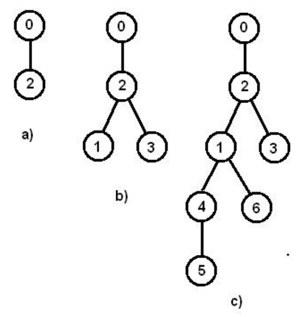

Fig. 3. Tree of obstacle-free paths for the example on Figure 2

in the ET^o point. The agents that are already making a recolection will not be able to respond to these messages.

The strategy followed in this negotiation consists on diminishing the number of agents of class MR^3 which participate in the sample recollection. This is made by taking the agents that have greater lifting capacity, for which the explorer agent acts like mediator. In this selection process the shipper agents, are sorted in decreasing order of their lifting capacity, and are seleced those with greater capacity until achieving the amount of shipper agents needed to collect the discovered samples.

This strategy has three basic purposes:

- To diminish the number of shipper agents which travel to a point of the environment. Doing this we can guarantee that having less agents we reduce the number of conflicts in crossing paths.
- To maximize the amount of collected samples because the agents are loaded at their full capacity.
- To diminish the amount of information about the environment that must be stored in the free-path tree discovered that is updated by each recollector agent who discovers a new path.

3.4 Path conflict resolution by a negotiation mechanism

The negotiation principle followed by the agents in our system tries to optimize the global objective that is to collect the greatest possible number of samples in the smallest period of time. Based on this principle, the negotiation between agents follows the next rules:

1. If an explorer agent mr^2_m and a shipper agent mr^3_k try to occupy the same $ET_{u,v}$ cell, the shipper has occupation priority over the explorer agent.
2. If two shipper agents mr^3_m and mr^3_k try to move to the same $ET_{u,v}$ cell, the shipper agent who is loaded and is going to deliver its load will have occupation priority. The agent who can not occupy the cell, begins a random walk and tries to recover its plan some movements later [1].

3. If two shipper agents mr^3_m and mr^3_k try to move to the same $ET_{u,v}$ cell, and are not loaded then the agent of greater lifting capacity will have ocupation priority over the other. The other enters into a state of random movements for recovering later his original trajectory. In the case that both agents have equal lifting capacity it will be decided randomly who will occupy the cell.
4. If two shipper agents mr^3_m and mr^3_k try to move to the same $ET_{u,v}$ cell, and both are loaded then the agent with the greater possible load will have priority over the other. If both have equal load capacity the decision of who has priority over the other will be at random. The agent whith less priority enters into a random movement state and tries to recover its path after certain number of movements [1].
5. If two explorer agents mr^2_m and mr^2_k try to move to the same $ET_{u,v}$ cell, then it will be decided randomly who will occupy the cell.

3.5 Experimental results under our cooperative model

The experiments were made for different proportions of explorer and shipper agents, going from a 10% to a 90% of explorer agents (increasing by 10% steps this amount) and for a 95% of explorer agents. Each one of these proportions was tested with different obstacle types. In Figure 4 we show the average time necessary to complete the 100% of the sample recollection, applied to different obstacle types and for each different explorer and recollector agent proportions. We can draw from Figure 4 the following conclusions:

1. Independently of the obstacle type, it can be observed that the time necessary to complete the task diminishes with the increase on the number of explorer agents until a value of 80% but it starts to increase again from this value which is observed for a 90% and 95%. Evidently more samples are discovered, but there are very few recollector agents to carry out them and these samples are left idle in the blackboard until a new opportunity appears.
2. The best proportion between explorer and recollector agents is between 70% and 80% of explorer agents.

Now we will compare the results obtained with our proposal against the results obtained using the two Wooldrige's models. Analyzing the Figure 5 it can be observed that the average time invested using our proposal to complete at 100% the task, with the different obstacle types, was significantly less than the average time under the Wooldridge's models. Moreover the worst results produced by our proposal (for the case of a 10% of explorer agents) were better than the results obtained using the Wooldridge's proposal.

4 Conclusions

1. Our proposal of agents with different capacities concerning observation capabilities as well as loading and displacement abilities, outperform the one that uses only one type of agent proposed by Wooldridge.

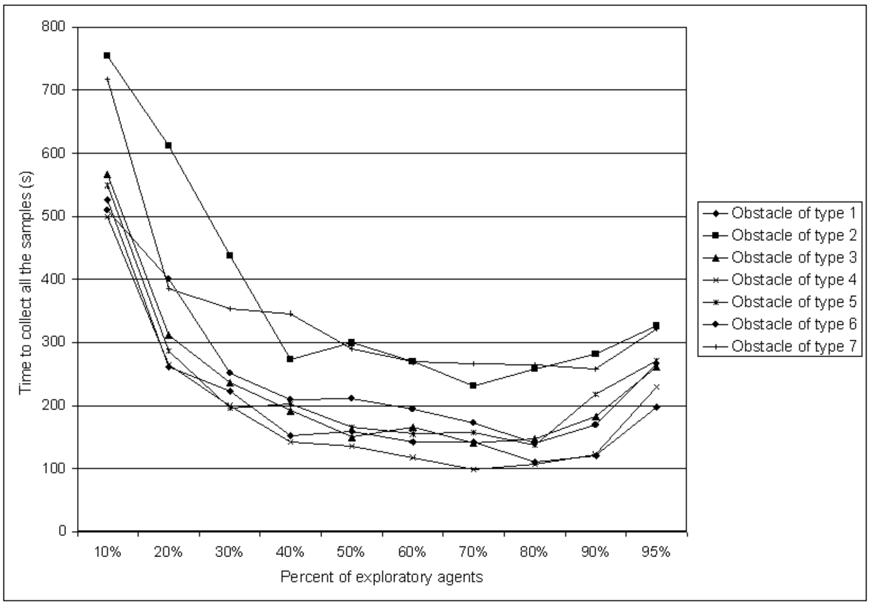

Fig. 4. Times to complete the recollection task with different proportions of explorer agents and different obstacle types

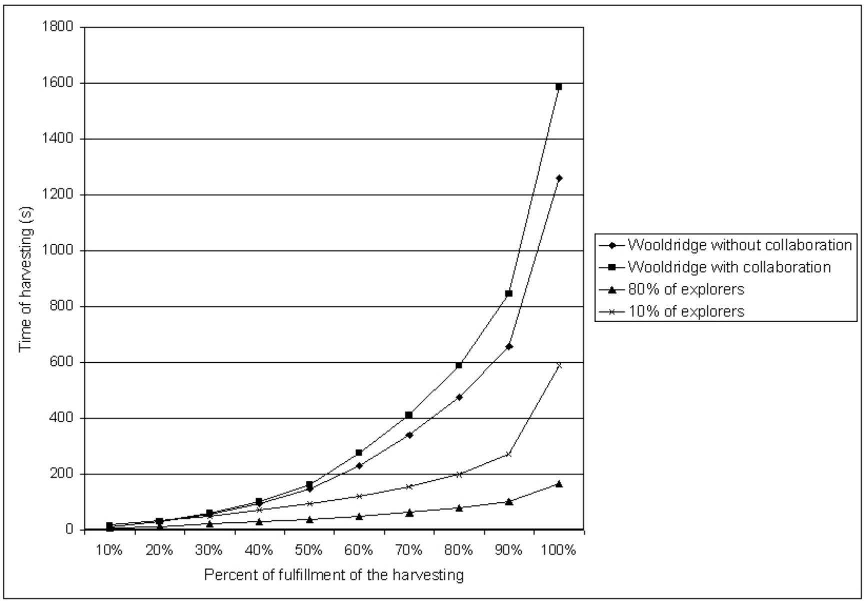

Fig. 5. Average time comparison between our model with the Wooldridge model for all the obstacle types

2. The negotiation and collaboration strategy for resolving conflicts, based on giving priority to shippers over the explorers, was quite efficient for these kind of problems.
3. The best performance of the system for the sample recollection, was obtained when using between 70% and 80% of explorer agents.
4. It has been experimentally shown that learning obstacle-free path method used in our simulator is a very efficient recognition form of the search space. In this sense, it must be mentioned that the size of the trees in most of the cases do not exceed a depth of three levels, and because of that the agents have a faster way to reach different points of the explored space.
5. The strategy of random movements of the shipper agents used to solve the problems of unexpected obstacles in their planned trajectories was quite effective, because noncollected samples never appeared.

References

1. Barraquand, J., Latombe, J.C.: Robot Motion Planning: A distributed representation approach. STAN-CS-89-1257 (1989) Stanford University.
2. Durfee, E.H.: Coordination of Distributed Problem Solvers. Kluwer (1988).
3. Finin, T., McKay, D., Fritzson, R.:An overview of KQML: A Knowledge Query and Manipulation Language. Technical Report (1992) U. of Maryland CS Departemnet.
4. Ginsberg, M.L.: Knowledge interchange format: the KIF of death. AI Magazine archive 12 (1991) 57–63.
5. Haddadi, A.: Communication and Cooperation in Agent Systems: A pragmatic Theory. Springer-Verlag (1996) Heidelberg.
6. Huhns, M.N., Singh, M.P.: Agenst and multiagents systems: Themes, approaches, and challenges. Distributed Artificial Intelligence 1–23 (1998). Morgan Kaufmann San Francisco CA.
7. Jennings, N.R.: Coordination Techniques for distributed Artificial Intelligence, In GMP O'Hare and N.R. Jennings, editors, Foundations of Distributed Artificial Intelligence (1996). 187–210 John Wiley and Sons Inc. New York.
8. Labrou, Y., Finin, T.: A Proposal for a new KQML Specification. http://www.csee.umbc.edu/kqml/papers/kqml97.pdf (1997).
9. Neches, R. Fikes, R., Finin, T. Gruber, T., Patil, R.,Senator, T., Swartout, W.: Enabling Technology for knowledge sharing AI Magazine 12(3) (1999) 36–56 Fall.
10. Rao, A.S., Georgeff, M.P.: An abstract architecture for rational agents. In C. Rich, W. Swartout and B. Nebel, editors, Proceeding of Knowledge Representation and Reasoning, Morgan Kaufmann (1992) 439–449.
11. Shoham, Y.: Agent-Oriented Programming, Artificial Intelligence 60-1 (1993) 51–92.
12. Wooldridge, M.: Intelligent Agents. In G. Weiss editor, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence MIT Press Cambridge MA (1999) 27–77.