Map Focus: a Way to Reconcile Reactivity and Deliberation in Multi-robot systems

S. Carpin [a], C. Ferrari [a], E. Pagello [a,b]

[a] *Intelligent Autonomous Systems Laboratory*
Department of Electronics and Informatics
The University of Padova, via Gradenigo 6/A I-35131 Padova, Italy

[b] *LADSEB-CNR*
Institute of Systems Science and Biomedical Engineering,
National Research Council, corso Stati Uniti 4, I-35127 Padova, Italy

Abstract

Cooperative multi-robot systems require both real-time responsiveness and some form of coordination to get the desired overall behavior. This can be obtained with a combined use of reactive and deliberative subsystems. In this paper we propose a novel technique for putting together these two components. The method is based on the idea that every robot maintains a local map and then dynamically focus its attention on the part which is relevant in the current context. The framework, which is fully distributed and scalable, is enriched with cooperative behaviors, i.e. behaviors pursued more than one robot. We provide the details of how the proposed idea has been studied in a simulated cooperative foraging task and proved to be effective.

Key words: Multi-robot Systems, Cooperative Robotics, Hybrid Architectures

1 Introduction

Nowadays multi-robot systems (MRS) seem to be one of the most promising domain for practical applications, but at the moment most of them are used only for research purposes. Indeed, we can think at MRS applications for space missions, service robotics, fire fighting, toxic waste management, and many other tasks. In such scenarios a team can be more suitable than a single robot, from the points of view of performance robustness and even cost. But MRS exhibit unique challenging aspects that require new paradigms in both the design and implementation stages, so that MRS research is not a straight extension of traditional single robot studies. A new class of problems arose and

has to be studied and resolved before we can see a massive use of MRS in the depicted areas. One of these concerns the use of deliberative and/or reactive components in such systems. In the past deliberative planning has been used in robotics but starting from the last eighties it has been mostly substituted by the *Behavior Based* approach, which proved to be more effective. But co-operative MRS, which are a particular instance of *Multi Agent Systems*, call for the use of deliberation, at least for the problem of coordinating distributed choices whose aim is to contribute to a common goal.
In this paper we propose a novel technique for balancing the two extremes of reactive behavior and social deliberation in MRS. The combined use of reaction and deliberation is needed because each approach give its own benefits which are not given by the other.

The paper is organized as follows: section 2 introduces the problem and outline related works, and in section 3 we discuss our proposal, called *Map Focus*. In 4 we give the details of we implemented Map Focus in a simulated cooperative foraging task, and in 5 we give the conclusions.

2 Multi-robot Systems, Deliberation and Reaction

At the end of the '80s a new class of systems was introduced, namely *Multi-robot Systems*. Sometimes it is stated that two or more robots performing in the same shared environment constitute a MRS, but we take a different point of view. In fact, according to the previous definition, a series of robotic arms in an assembly line constitute a MRS. In this work we consider MRS as autonomous mobile robots performing in a common shared environment not necessarily engineered for them. Such systems are suitable in many situations where a single mobile robot can not accomplish a hard task, like for example transportation of heavy and/or big objects, building surveillance, or disaster rescue ([2], [23]). Moreover MRS can be used in a variety of tasks where fault tolerance is a must. In MRS fault tolerance is gained thanks to the redundancy introduced, thus a great number of application fields could benefit from MRS. But MRS are not a straightforward extensions of traditional single robot systems, because of the great number of degrees of freedom which contributes to define the whole MRS. [13] and [17] outline a number of possible different arrangements that can be obtained in the MRS architecture and [27] illustrates the challenges of MRS design and implementation.
We focus our attention on *cooperative* MRS, i.e. systems where each element aims to the same goal, so that success or failure will concern the whole team. Cooperation can be obtained in various way. It is possible to take a minimalist approach and design simple agents governed by a small set of rules that, without being aware of other members, work together toward a common goal ([19], [35]). This *swarm effect* design approach come from biological and etho-

logical evidences and from cybernetics ([10]), and is being widely studied, even if clear rules and designed methodologies are still not identified (see for examples [7], [28],[32]). Cooperation can also be obtained through *coordination*. In this case every robot has a certain degree of knowledge that it is member of a group with a shared goal. So, robots can deal with models of other robots, like geometric models for safe navigation or models of other robot actions and intentions to take local decisions. This problem is closely related to those found in Distributed Artificial Intelligence and has received great attention so that a number interesting solutions have been proposed (see [5], [8], [9], [16],[21], [22], [26] and [30], while [33] provides a survey on coordination/cooperation techniques). Moreover we focus on cooperative *Distributed Autonomous Robotic Systems*, i.e. MRS which lack a centralized controller. This choice is driven to overcome the weakness of centralized solutions, where a failure of the central controller lead the whole system to fail.
Even if MRS came from traditional single robot systems, it soon become evident that new challenging issues were to be addressed ([27]). One point which is still being investigated is how to balance reactivity and deliberation in multi-robot systems.

2.1 Reactivity and Deliberation

Deliberative planning played an important role in the early days of autonomous robotics and a number of autonomous robots has been designed and implemented using this approach ([18],[29]). However, the so called *sense-think-act* paradigm proved to be not suitable under real time constraints imposed by operating in dynamical environments. The seminal work by Brooks ([11]) introduced the novel *Behavior Based* approach, which proved to be very effective. According to Brooks an intelligent behavior by a situated agent can be achieved without an explicit model of the environment ([12]). After this design revolution, hybrid architectures were introduced, i.e. frameworks with deliberative and reactive subsystems (see for example [1], [3], [4]), with the aim of putting together deep reasoning capabilities together with real time responsiveness, so that planning continued to play a role in robotics ([24]).
The advent of MRS introduced new problems. In fact every robot of a MRS operates in a highly dynamic environment, because of the sudden and impredictable changings introduced by other robots. This would call for a pure reactive approach, but this is not the case. In fact with a pure reactive handling of this overloaded input of perceptual stimuli there is the risk of getting a *skizophrenic behavior*, with a frequent and useless triggering of different behavior. Moreover, a pure reactive approach poorly adresses the problem of coping with strictly coupled tasks, which in turn is usual in cooperative robotics. So it seems reasonable to introduce some kind of negotiation between robots, as is often done in multi agent systems, and in fact a number of researchers started

to consider this problem ([20]), particularly for the case of multi agent systems, but at the moment there is not a widely accepted solution or paradigm.

3 Map Focus

Our proposal addresses cooperative distributed autonomous robotic systems. This means that the systems we are interested to are composed by a number (two or more) of robots, each one with its own computing capabilities and control systems. Moreover each unit is autonomous, i.e. operates without the help of any external entity, so that each robot is equipped with sensors and actuators to accomplish the given task. Finally, robots cooperate to reach a common (*shared*) task.
We devised three key issues which was assumed in our research:

- *Awareness*: each robot is member of a team and is aware of this. Despite minimalist approaches, each robot knows that its individual goals are directed towards a common team goal and that its choices and actions will influence other robots' behavior. For this reason robot choices has to match possibly conflicting individual and team goals
- *Locality*: each robot operates in a dynamical initially unknown environment and has bounded computational capabilities. This means that it will not be able to process all the information coming from sensors. So, to correctly operate it has to choose which information to use. It is reasonable to assume that it will handle only local information, and this calls for a criterion for deciding when information can be considered local
- *Active Sensing*: sensing is a time consuming activity, and this has to be coupled with real time constrains. This means that a robot can not spend too much time in sensing, because this could not be matched with the environment time scale. Thus, cooperation should be used also in the sensing process, by mean of some sort of communication. However, communication should be compliant with the goal of scalability and distributed operation

On the basis of the former assumption we developed a novel architecture, whose general schema is depicted in figure 1. Two subsystems, one deliberative and one reactive operate in parallel to get the desired behavior. Such subsystems are coupled by mean of a third subsystem, named Map Focuser, an active entity which acts as a filter between the two. Each subsystem will now be discussed. It has to be noted that while some other authors opt for a not well defined separation between reactivity and deliberation ([6],[34]), in our implementation the two subsystems are clearly divided. This choice cames from the observation that in this way we can easily reuse classical techniques used for reactivity and deliberation.

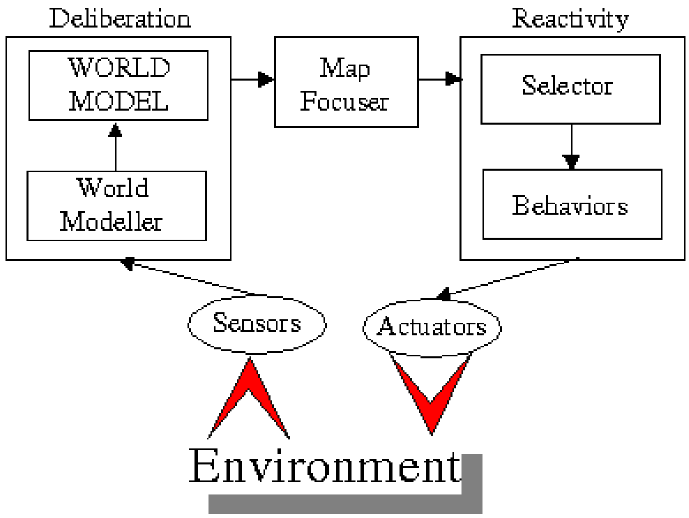

Fig. 1. Overall schema

3.1 World Modeler and the Deliberative Subsystem

"World Modeler" handles the world model. This is the only subsystem which modifies it, inserting, updating or deleting objects. The world model can be organized in any way. It can be a set of logic statements, or of geometric objects. World modeler reads sensor inputs and, also on the basis of the current state of the world model, accordingly updates the world model. Since real time constrains are a must, sensor inputs have to be read and processed in parallel. In this way decisions based on critical sensor readings, like for example a laser beam which indicates a collision danger, are not postponed until a time consuming interpretation of a digital image is carried out.

3.2 Map Focuser

"Map Focuser" is a separate active entity which operates on the world model handled by "World Modeler" and outputs a simplified version for the subsequent functional block. Since it only reads the world model, there is no problem of concurrency interference between Map Focuser and World Modeler. Map focuser produces a local version of the world model, i.e. a subset of objects considered local to the robot. This module addresses the "locality" issue previously outlined. To accomplish its task, Map Focuser needs a metric to decide for every object in the world model if it is local, i.e. if it has to be inserted in the focused map. Such metric is not necessary a geometric distance, because world model can also contain non geometric objects, like for example robot intentions or messages (see section 4 for an applicative example of this concept).

3.3 Selector, Behaviors and the Reactive Subsystem

The reactive subsystem is composed by two logic devices: a selection mechanism ("Selector") and a set of behaviors. Selector takes as input the focused map produced by Map Focuser and sets a set of boolean conditions which will trigger the appropriate behavior from the behavior set.

3.4 Group Behavior

Coordinated behaviors between two or more robots are obtained through the combined use of reaction and deliberation. This because while every robot has its own behavior selector mechanism, the conditions which feed selector come from deliberation, which, relying on the awareness hypothesis, takes into account other team members. For this reason we devise a three layers behaviors set. Low level behaviors set (LLBs) implements basic robot capabilites (like wandering, avoiding obstaclese, etc.). Built on the top of LLBs, High Level Behaviors (HLBs) get more sophisticated actions (safe wandering, etc.). LLBs and HLBs are robot level entities, while Cooperative Behaviors (CBs) are executed by two or more robots, i.e. they are executed when a set of conditions shared between robots is verified. Social deliberation helps to determine when such conditions are verified. In the example later introduced (section 4) this is gained through explicit communication. Figure 2 shows an example of the three layers architecture (HLBs are on the left, LLBs are on the right and the CB *Homing With Puck* is in the middle).

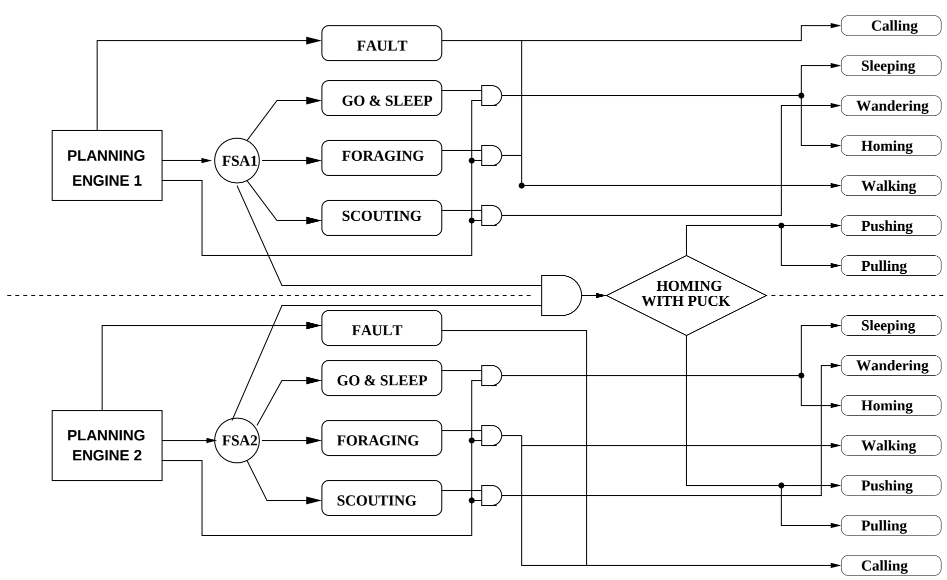

Fig. 2. CBs, HLBs and LLBs (taken from the case study presented in section 4)

Balancing reactivity and deliberation is obtained thanks to the “Map Focuser”. As previously stated, one of the problems of a pure reactive approach for MRS is that an overloaded flow of input information can result in a too frequent switching between different behaviors. The reduced (focused) map produced by “Map Focuser” addresses this problem. Cutting away non local object from the world model helps getting a reduced flow of sensor inputs for behavior selection. On the other hand, since the deliberative subsystem parallel querys every sensor and inteprets its data also on the basis of the whole map, it is able to get high level interpretation from low level data. So, if Map Focuser performs a strong focus on the map, i.e. it cuts away many objects, systems' behavior is mostly deliberative, since selector operates only on a few objects which came from world modeller. But if Map Focuser performs a poor focus, Selector has access to a wide amount of information and can then exhibhit a more reactive behavior (see figure 3)

Fig. 3. The role of Map Focuser

Finally we have to address Awareness and Active Sensing. This can be done using some sort of communication, maybe implicit ([25]), or explicit. In the current context we deal only with explicit communication, thought as message passing or broadcasting. A communication device is seen as a kind of sensor and its data (i.e. messages) are processed by World Modeler. In this way it is possible to include in the model not only physical objects (like obstacles and other robots), but also other robot intentions (awareness). Moreover messages can contain information about the environment which can be added to the local world model even if the robot has not perceived it. Furthermore, a robot can ask for information from other robots, which can answer if they are able to do so (active sensing). This exchange of information message fits in the focus framework, because critique messages can be kept by the Map Focuser and are then given as input to the behavior selector (see section 4 for an application of these ideas).
It has also to be outlined that this *block schema* applies both to homogeneous and heterogeneous systems because every robot holds its own map and can build it according to his sensors, so that different robots can build different kind of maps and filters.

4 Cooperative Foraging

We developed a simulated framework to test the effectiveness of the Map Focus idea. The considered task is *Cooperative Foraging*: a team of robot performing in a common area is required to find scattered items and to bring them back to a home area. This task is inherently cooperative and is commonly used as a testbed for coordination techniques ([26],[31]). We introduce one more constraint: two robots are necessary to move a puck. This means that a robot is not allowed to bring an item in the home area if it is not supported by a mate which follows him and explicitly communicate that it is supporting him. The simulated framework has been developed using VLAB, our general purpose multirobot simulator ([14]). Figure 4 illustrates an example of the redering yield by the VLAB simulation environment devoted to cooperative foraging. The simulated MRS is composed by a team of heterogeneous robots,

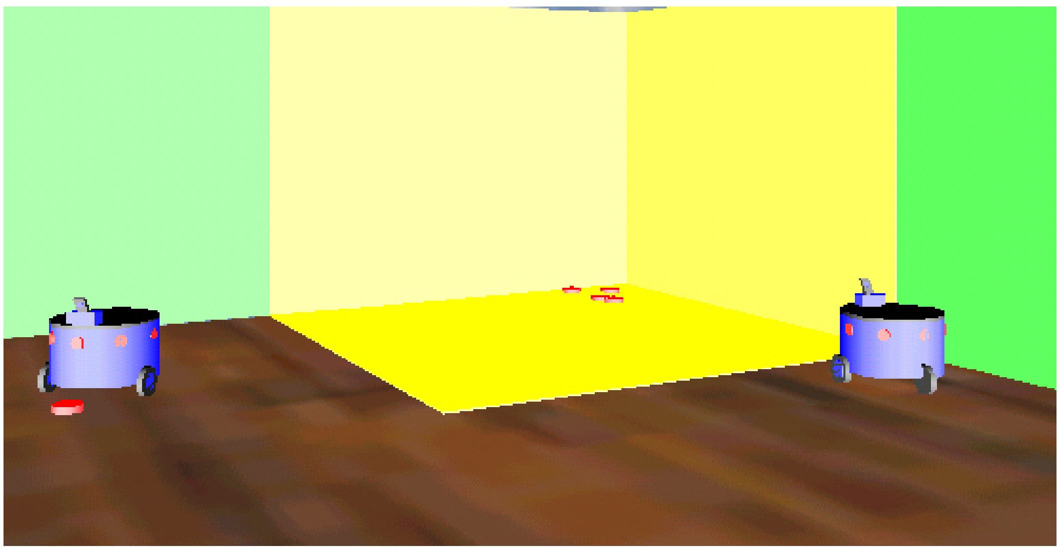

Fig. 4. VLAB rendering of simulated foraging

each one equipped with:

- a gripper device to pick up items; the gripper is also equipped with a sensor which allows to know if something has been caught
- a communication device which allows broadcast and unicast communications
- a GPS-like device, which tells each robot its position (relative to a room based coordinate system)
- 12 range scanners disposed on the perimeter which measures the distance of the other robots (but not of the walls)
- a camera and a frame grabber

Robots operate in an indoor $100m^2$ square area with the home area placed in a corner. At starting time robots do not know their positions nor pucks and home locations, so they only rely on sensed information.

4.1 *Implementing "Map Focuser"*

The Map Focuser module is implemented as a set of concurrent threads, each one operating on a different sensor. Each thread can produce geometric objects which are inserted in the World Map. Each geometric object is located in the generalized space *GS, where* GS is defined as [1]

$$GS = \mathbb{R}^3 \bigcup \{EveryWhere\}$$

Everywhere is a symbol added to the Euclidean space $\mathbb{R}^3$ and its function will be soon explained. The map is organized as a set, so there is not an order between the objects. The space *GS* is introduced because the map keeps not only the model of physical objects (like robots or items) which has a physical location in the Euclidean space, but also the models of the intentions of the agents and of the information they are exchanging. Such entities are not located in the precise point of the space, but rather affect the whole environment. For example if a robot asks for some information, this fact will be modeled as an object of type "Communication" whose location is *EveryWhere*, so every other robot should consider it as relevant, i.e. local. The map is owned and managed by the World Modeler. This means that inserting and deleting on and from the set is possible only to that module. Other modules may only read the contents of the set, but can not modify it.

4.2 *Implementing "Map Focuser"*

Thanks to the choices made for the map, the Map Focuser is extremely simple: it is a thread that continuously scans the set of objects which constitutes the map and outputs a subset of it. An element belongs to the output subset if is considered local to the robot.

Algorithm 1, which shows how the focus is obtained, is based on some routines:

- *SerialCopy* gives a serialized copy of the map, so that it is possible to sequentially scan it (remember that the map is generated as a unordered set of objects)

[1] since we do not deal with flying objects, GS could also be defined as $\mathbb{R}^2 \bigcup \{EveryWhere\}$. However the given definition does not raise any computational complication, so we think to objects as entities in $\mathbb{R}^3$

Algorithm 1 Algorithm for building the focused map

```
INPUT Map: the global map
OUTPUT FocusedMap: the focused version of the global map
loop
  MapCopy ← SerialCopy(Map)
  for j=0 to | MapCopy | do
    Obj ← MapCopy[j]
    local ← IsLocal(Obj,CurrentState)
    if In(FocusedMap,Obj) AND local then
      Update(FocusedMap,Obj)
    else if In(FocusedMap,Obj) AND NOT local then
      Delete(FocusedMap,Obj)
    else if local then
      Insert(FocusedMap,Obj)
    end if
  end for
end loop
```

- *IsLocal* is a boolean function which determines if the object has to be inserted in the focused map; this selection is based on the generalized position of the object and on the state of the system
- *In* is a boolean function which tells if the object is present in the focused map
- *Insert*, *Delete*, and *Update* perform the corresponding operations on the focused map

The routine which performs the focus on the map is extremely simple and can be implemented in a very efficient way, so that newly introduced geometric objects are processed in a short time. Figure 5 shows an example of the performed filtering on the map.

4.3 Behavior Selection

Behavior selection is implemented as a Finite State Automaton and is based on the focused map produced by Map Focuser. Such map is scanned and a set of relevant boolean conditions is set. On the basis of verified conditions and of the current state of the selector one behavior is triggered in a winner-take-all basis. In the simulated implementation behavior selection explicitly considered other robot possibly choice, using a game theory based framework (see [15] for details).

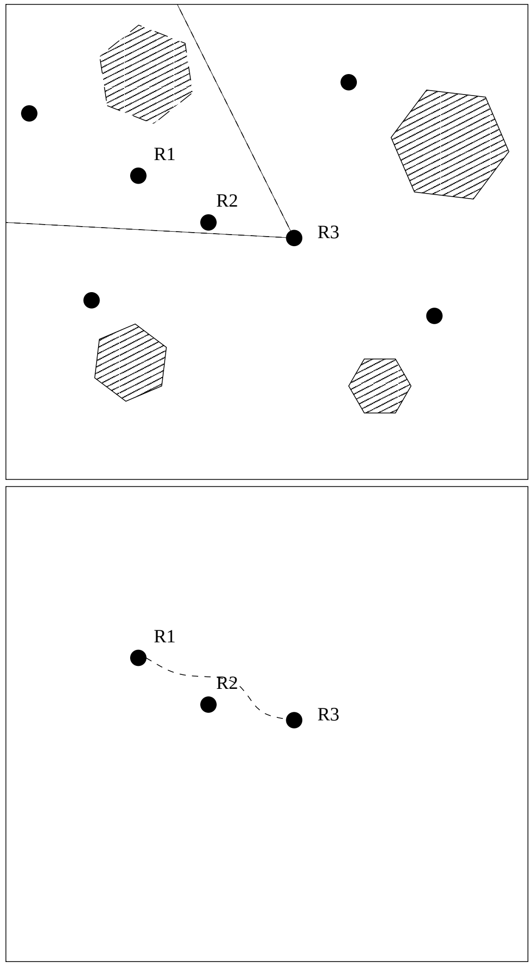

Fig. 5. An example of map focus

4.4 The Behavior Set

The set of available LLBs implemented is the following:

- *Wander*: randomly move in the environment
- *GoToItem*: move toward a recognized item
- *CatchItem*: grasp an item
- *Call*: send a message requiring help to bring an item in the home area
- *WaitForAMate*: if a previous request has been positively answered, wait for the incoming mate
- *GoToHome*: move toward the home area grasping an item
- *WaitForHomeFree*: if other robots are in the home area, wait outside
- *EnterHomeArea*: enter in the home area
- *AnswerYes*: positively answer to a help request
- *AnswerNo*: negatively answer to ha help request
- *GoToMate*: reach a robot

- *Support*: move towards the home area supporting a robot which holds an item (remember that two robots area needed to bring an item to the home area)

Based on the shown LLBs, two HLBs has been implemented:

- *GET-ITEM*: move in the environment, find an item and get it
- *NEGOTIATE*: ask for help and wait or answer and eventually move to the mate

The only CB is *BRING-ITEM-TO-HOME*, which is performed in a different way by the leading robot and by the supporting robot. The leading robot carries the item, while the supporting robot follows him, after having explicitly declared that it will give such support.

4.5 Experimental results

The aim of the simulation was twofold. First it was necessary to verify the effectiveness of the focus idea. To do this, we run two set of trials, the first one with the map focuser on and the second with the map focuser off (to switch of the map focuser we modified the *IsLocal* routine, so that it returns always true, and then no object is cut off the map). Figure 6 compares the times required to bring the first element in the home area. Every point in the figure is the average of 5 tests, to smooth out the effects of the random distribution of the elements in the home area.

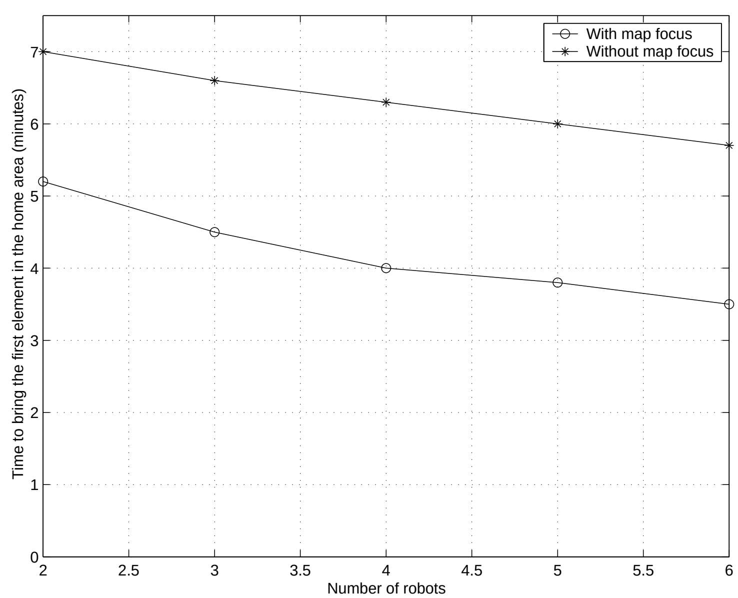

Fig. 6. Performance with *Map Focuesr* on and off

It can be noticed that not only the focus yield lower times, but also the decrease is more evident while augmenting the number of robots in the team. The second goal was to verify that the use of the focus framework enables the multirobot team to get an increase of the performance with the increase of the team size. Figure 7 plots the time required to complete the mission for different team sizes.

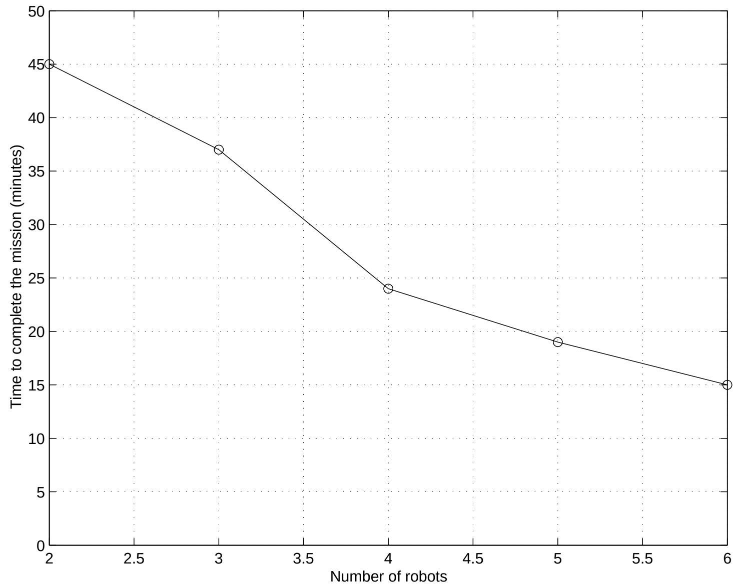

Fig. 7. Team performance

The trend cleary shows that performance increases with team size, thanks to the coordination framework introduced. It can also be observed that even sized teams perform better then odd. This because two robots are necessary to carry an item, and in some situations in odd sized teams, a robot has to wait for a supporting mate while they are all already matched, a situation less likely in a even size time.

5 Conclusions

We discussed the major problems in the balancing reactivity and deliberation in multi-robot systems and introduced a novel technique for balancing these two components. The method, based on a procedure called "Map Focus" allows an easily tuning between pure reaction and deep social deliberation. The proposed idea, which apply both to homogeneous and heterogeneous systems, has been implemented in a simulated cooperative foraging task and proved to be effective. We are currently working to use this framework for real multi-robot systems.

Acknowledgements

This research has been partially supported by the Ministry for Education and Research (MURST Certamen project), by the Italian National Council of Research (CNR special project), and by the Italian Energy Agency (ENEA Parallel Computing Project).
The authors would like to thank Pierluigi Patuelli and Federico Montesello for their contribution to this research.

References

[1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *The international Journal of Robotics Research*, 17(4):315–337, 1998.

[2] T. Arai and J. Ota. Let-us work together - task planning of multiple mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 298–303, 1995.

[3] R. Arkin. Path planning for a vision-based autonomous robot. In *Proceedings of the SPIE Conference on Mobile Robots*, pages 240–249, 1986.

[4] R. Arkin. Integrating behavioural, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105–122, 1990.

[5] H. Asama, K. Ozaki, A. Matsumoto, Y. Ishida, and I. Endo. Development od task assignment system using communication for multiple autonomous robots. *Journal of Robotics and Mechatronics*, 4:122–127, 1992.

[6] M.S. Atkin, D.L. Westbrook, and P.R. Cohen. Hac: A unified view of reactive and deliberative activity. In M. Hannebauer, J. Wendler, and E. Pagello, editors, *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, Lecture Notes in Computer Science. Springer, 2001.

[7] G. Beni and J. Wang. On cyclic cellular robotic systems. In *Japan-USA Symposium on Flexible Automation*, pages 1077–1083, 1990.

[8] S.C. Botelho and R. Alami. Cooperative plan enhancement in multi-robot context. In E. Pagello et al., editor, *Intelligent Autonomous Systems 6*. IOS Press, 2000.

[9] S.C. Botelho and R. Alami. A multi-robot cooperative achievement system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2716–2721, 2000.

[10] V. Braitenberg. *Veichles: Experiments in Synthetic Psychology.* MIT Press, 1984.

[11] R.A. Brooks. A robust layered control systems for mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, 1986.

[12] R.A. Brooks. Intelligence without reason. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 569–595, 1991.

[13] U. Cao, A.S. Fukunaga, and A.B. Kahng. Cooperative mobile robots: Antecedents and directions. *Autonomous Robots*, 4(1):7–27, 1997.

[14] S. Carpin, C. Ferrari, and E. Pagello. A framework for distributed simulation of multirobot systems: the vlab experience. In L.E. Parker, G. Bekey, and J. Barhen, editors, *Distributed Autonomous Robotic Systems 4*, pages 45–54. Springer, 2000.

[15] S. Carpin, C. Ferrari, E. Pagello, and P. Patuelli. Bridging deliberation and reactivity in cooperative multi-robot systems through map focus. In M. Hannebauer, J. Wendler, and E. Pagello, editors, *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, Lecture Notes on Computer Science. Springer, 2001.

[16] L. Chaimowicz, Thomas Sugar, Vijay Kumar, and Mario F. M. Campos. An architecture for tightly coupled multi-robot cooperation. In *Proceedings og the IEEE International Conference of Robotics and Automation*, pages 2992–2997, 2001.

[17] G. Dudek, M.R.M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397, 1996.

[18] R. Fikes and N. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[19] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss. Self organaizing robots based on cell structures - cebot. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*, pages 145–150, 1988.

[20] L. Iocchi, D. Nardi, and M. Salerno. Reactivity and deliberation: a survey on multi-robot systems. In M. Hannebauer, J. Wendler, and E. Pagello, editors, *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, Lecture Notes in Computer Science. Springer, 2001.

[21] D. Jung and A. Zelinsky. An architecture for distributed cooperative planning in a behavior-based multi-robot system. *Robotics and Autonomous Systems*, 26:149–174, 1999.

[22] O. Khatib, S. Quinlan, and D. Williams. Robot planning and control. *Robotics and Autonomous Systems*, 21:249–261, 1997.

[23] H. Kitano and S. Tadokoro. Robocup rescue. a grand challenge for multiagent and intelligent systems. *AI Magazine*, 22(1):39–52, 2001.

[24] D. McDermott. Robot planning. *AI Magazine*, pages 55–79, Summer 1992.

[25] E. Pagello, A. D'Angelo, F. Montesello, F. Garelli, and C. Ferrari. Cooperative behaviors in multi-robot systems through implicit communication. *Robotics and Autonomous Systems*, 29(1):65–77, 1999.

[26] L.E. Parker. *Heterogeneous Multi-Robot Cooperation.* PhD thesis, Massachusetts Institute of Technology, January 1994.

[27] L.E. Parker. Current state of the art in distributed autonomous mobile robots. In L.E. Parker, G. Bekey, and J.Barhen, editors, *Distributed Autonomous Robotic Systems 4*, pages 3–12. Springer, 2000.

[28] R. Pfeifer. Building fungus eaters: Design principles of autonomous agents. In *Simulation of Adaptive Behavior*, pages 3–12, 1996.

[29] E. Sacerdoti. *A structure for plans and behavior.* American Elsevier, 1977.

[30] A. Saffiotti and N.B. Zumel. Multi-robot team coordination using desirabilities. In E. Pagello et al., editor, *Intelligent Autonomous Systems 6*, pages 107–114, 2000.

[31] M. Schneider-Fontán and M.J. Matarić. Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 14(5):815–822, 1998.

[32] L. Steels. Exploiting analogical representation. In P. Maes, editor, *Designing Autonomous Agents*, pages 71–88. MIT Press, 1990.

[33] E. Todt, G. Raush, and R. Suàrez. Analysis and classification of multiple robot coordination methods. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3158–3163, 2000.

[34] G. Weiss. Cognition, sociability, and constrains. In M. Hannebauer, J. Wendler, and E. Pagello, editors, *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, Lecture Notes in Computer Science. Springer, 2001.

[35] B. B. Werger. Cooperation without deliberation: A minimal behavior-based approach to multi-robot systens. *Artificial Intelligence*, 110:293–320, 1999.