An Architecture for Reactive Cooperation of Mobile Distributed Robots

O. Simonin, A. Liégeois, and P. Rongier

LIRMM - Université Montpellier II,
161 rue Ada, 34392 Montpellier Cedex 5, France
{simonin,liegeois,rongier}@lirmm.fr

Abstract. This paper shows how to include a crude but efficient communication component into reactive robotic multi-agent systems. The basis of the work relies on the concept of self-satisfaction of an agent composed of a selfish tendency and an altruistic one. A signal, representing the variation of an agent's "interactive satisfaction", is emitted in its neighborhood. It measures the expected efficiency of cooperation. The information propagates from agent to agent within a certain range. It is shown how the information is converted into a vector able to alter instantaneously the agent's motion when added to the classical vectors which generate the selfish motion: move-to-goal and avoid-obstacle. Then the paper presents the architecture for implementing such mechanisms to achieve robust adaptive cooperation. The results of simulation experiments are presented which demonstrate the efficiency of the generalized dynamic potential field including communication of satisfaction. In particular, the method allows the agents to escape from a deadlock situation, and makes the collective behavior converge towards a steady flow of traffic.
Keywords: multi-agent systems, cooperating mobile robots, robot communication, reactive architecture, self-organizing robots

1 Introduction

In the field of distributed autonomous systems, cooperation appears to be one of the most important way to perform meaningful tasks.

As it is emphasized in [11], many robotic applications are inherently distributed in space, time, or functionality, thus requiring distributed solutions. Adaptive cooperation techniques allow to handle repetitive tasks or tasks which need cooperation of several robots for their completion [3][11].

Many animats which have been implemented with reactive behavioral control techniques have only local interactions with the environment and the other agents. Nevertheless, they can perform complex foraging tasks [4][6][12], multi-robot navigation [1], cooperation tasks [6][9], and distributed problem solving [7].

However, the poor level of communication and cognition of reactive architectures cannot so far handle easily goal-driven behavior [10] and does not allow explicit cooperation [4] to perform complex tasks. The aim of this work is to include a robust but simple intentional cooperation into a reactive architecture without degrading its intrinsic qualities.

In a previous work [13], we have introduced the self satisfaction which expresses the performance of the agent's current task, and the interactive satisfaction which results from agent interactions. This last information is broadcast as a simple signal in order to be directly converted by agents into vectors which may be combined to other vectors (goal attraction, obstacle avoidance).

This paper concentrates on the process of satisfaction propagation and shows that it contributes to adaptive cooperation, while avoiding deadlocks ("intelligent behavior").

The paper is organized as follows: Section 2 presents the model of communication and the related altruistic behavior vector. We then describe the cooperative architecture in section 3. Section 4 shows some typical simulations and discusses experimental results.

2 From Satisfaction to Altruism

This section presents a formal description of the components of an agent's satisfaction, and how they are converted into vectors consistent with those resulting from other needs: avoiding collisions and reaching a destination.

2.1 Cooperation and Satisfaction

The satisfaction approach. From the designer's point of view, the agent behavior must satisfy *goal-oriented*, *conservative* and *cooperative* functions. The principle of our agent model relies on the maximization of agent satisfactions. At any time, the agent tries to maximize

- either its selfish interests: by selecting the optimal task and by emitting repulsive signals to agents which hinder its work or attractive signals to altruistic agents which can help it,
- or the collective interests: by helping other agents, i.e. by reading agent signals and then computing altruistic behaviors.

Sub-satisfactions. As introduced in [13], we distinguish the personal satisfaction, the interactive satisfaction and the empathy satisfaction.

The *personal satisfaction* P measures the progress of the agent's task. The *empathy satisfaction* E is the average value of personal satisfaction of its acquaintances, it expresses the *altruism* of the agent. The instantaneous satisfaction of an agent i at time t is $Sat_i(t) = (1-\alpha).P_i(t) + \alpha.E_i(t)$, where α is the *altruistic factor* of the agent ($\alpha \in [0,1]$).

The *interactive satisfaction* I results from agent interactions. An agent computes the interactive satisfaction as a reaction or an intention to other agents.

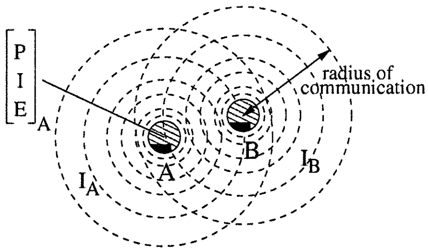

Fig. 1. Broadcasting of agent signals

Signals of Interactive Satisfaction. Each agent i broadcasts only its interactive satisfaction $I_i(t)$ because it is an intentional signal to alter agent interactions. This signal is broadcast within a bounded distance from the agent (Fig. 1). The *intensity* and the *variation* of the signal define the meaning of the communication.

2.2 Altruistic Behavior

Each agent alters its selfish behavior by processing the signals received from its neighbors, that generates a component of its velocity vector.

Agent-agent altruism: Agents use interactive satisfaction either to emit a repulsive signal ($\Delta I\searrow$) or to emit an attractive call ($\Delta I\nearrow$). The motion behavior clearly depends on the other agent interactive satisfaction variation $\Delta I_A(t)_T = I_A(t) - I_A(t-T)$.

Thus, in the vector model, this cooperative behavior is defined with the *altruism vector* of an agent (B) relative to another agent (A) $\overrightarrow{\vartheta_{B/A}}(t)$:

$$\overrightarrow{\vartheta_{B/A}}(t) = k.S(\Delta I_A(t)_T).\frac{\mid I_A(t) \mid}{\left\|\overrightarrow{AB}\right\|^2}.\overrightarrow{BA} \tag{1}$$

$$S(x(t)) = \begin{cases} Sign(x(t)) & if\, x(t) \neq 0 \\ Sign(x(t-T)) & if\, x(t) = 0 \ (to\, ensure\, continuity) \end{cases}$$

Equation 1, where k is a gain, defines a force field. The interest of equation 1 lies in the fact that it can be used for various situations: i) just one agent moves ii) both agents move iii) none of the agents move but they notice significant interactive satisfactions.

Multi-agent altruism: A situated agent close to others can simultaneously perceive several attractions and repulsions. In order to define a new altruism vector we split the multi-agent problem into a set of agent-agent problems. The altruistic reaction of an agent (B) close to N acquaintances noted Λ is computed as follows

$$\overrightarrow{\vartheta_{B/\Lambda}}(t) = \sum_{j\in\Lambda} \overrightarrow{\vartheta_{B/j}}(t) = \sum_{j\in\Lambda} k.S(\Delta I_j(t)_T).\frac{|I_j(t)|}{\left\|\overrightarrow{BJ}\right\|^2}.\overrightarrow{BJ} \tag{2}$$

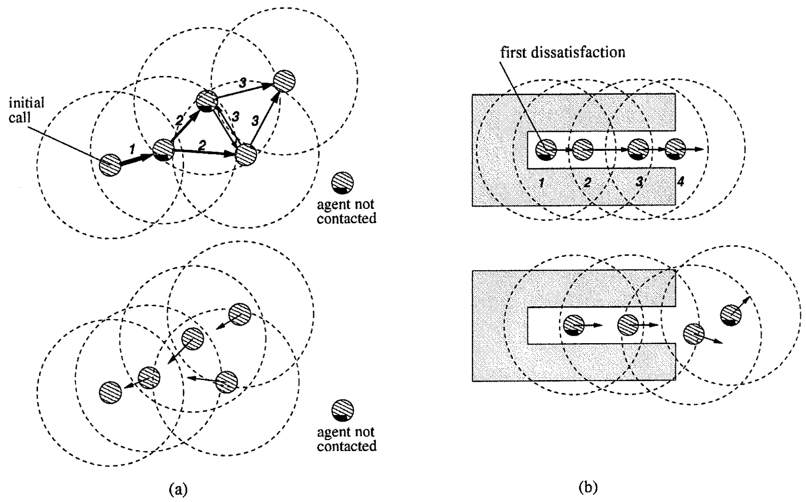

Fig. 2. (a) Intentional propagation for attraction (b) Implicit propagation of repulsion (the arrows are numbered following the sequence of emitted signals)

2.3 Sequential Signal-Passing

Sequential passing of attractions: When an agent perceives an increasing signal, it may decide to follow the call. In this case, the agent also emits an attractive signal in order to call its own acquaintances (implicit recruitment). The force of the signal variation decreases during the propagation so as to limit the number of attracted agents (see. Fig. 2.(a)).
Sequential passing of repulsions: If an agent is immobilized by another, it can emit a dissatisfaction signal to query a movement of the hindering agent. But, if this other agent is also immobilized, a chain process of dissatisfaction emission is performed until an agent is able to move (Fig. 2.(b)). This implicit passing of dissatisfaction to all involved agents releases the deadlock situation.

3 Principle of the Architecture

Our architecture is based on a classical schema-based reactive control [4] using vector combination [1][8][14]. Actions are selected and computed from proprioceptive information (energy, conservative functions) and exteroceptive information (obstacles, goals), with the addition of the new vector called *altruistic vector*. It represents the cooperative part of the agent's behavior.

3.1 Primitive Behavior

Each task (primitive behavior) is an *action* which may be triggered by an internal or environmental *stimulus*. The test to release an action is defined by a set of boolean conditions on perceptions, *Cond(task)*. Moreover, each condition induces a measure of intensity computed from perceptions and called *Int(task)* (defined on the interval $[0, 1]$).

Some tasks are *functions for survival*, i.e. they are useful for robot working. These tasks have priority on others (with a subsumption control [5]).

3.2 Algorithm of Action Selection and Combination

Data : $task_c$, P, I, Sensors
if *survival function triggered* **then**
  if *motion task* **then** compute $\overrightarrow{V}_{goal}$
end
else
  $task_m \leftarrow \{\forall i \; Int(task_m) \geq Int(task_i) \text{ and } Cond(task_m) = true\}$
  if $Int(task_m) > P$ **then** $task_n$=$task_m$; F=Int($task_m$)
  else $task_n$=$task_c$; F=P
  Read signals: I_1, I_2,..
  if $task_c$ *is altruism and* $sign(\Delta I_c(t)) = sign(\Delta I_c(t - T))$ ***then***
   continue $task_c$
  else
    I_{max} =$Max(|I_1|, |I_2|,..)$
    if $\alpha.|I_{max}| > (1 - \alpha).F$ ***then*** $\overrightarrow{V}_{goal} = \overrightarrow{\vartheta}_{I_{max}}$
    else
      $task_c$=$task_n$
      if $task_c$ is a motion ***then*** $\overrightarrow{V}_{goal} = \overrightarrow{V}_{task_c}$
    end
  end
end
if *moving action* **then**
  Integrate signal repulsions :
  $R = \{ \bigcup I_i / \Delta I_i < 0\}$; $\overrightarrow{V}_{alt} = \sum_R \overrightarrow{\vartheta}_{R_i}$
  Integrate obstacles sliding :
  Compute $\overrightarrow{V}_{obs}$ for avoiding obstacles
end
if *moving action* **then** Compute $\overrightarrow{V} = \gamma_1.\overrightarrow{V}_{goal} + \gamma_2.\overrightarrow{V}_{obs} + \gamma_3.\overrightarrow{V}_{alt}$
Perform behavior

Algorithm 1: Generic Algorithm of Altruistic Robots

Task Selection. If no survival function is released, the algorithm (Alg. 1) selects the next task to perform (a new one or the current one).

First, the task which has the maximum perceived intensity among the triggered tasks is computed (line 2). If this task has an intensity greater than the current task performance index (= personal satisfaction P) it becomes the new candidate task, noted $task_n$ (line 3). But this task may be replaced by an altruistic behavior. For this purpose, we compute the intensity F of $task_n$ selection (line 3).

Altruistic Behavior Selection. If the agent is already performing an altruistic task and the relative perceived signal (I_c) keeps the same sign of variation, it keeps the same behavior (line 4).

Otherwhise, as the agent perceives interactive satisfactions I from others, it can reply to those signals by exhibiting an appropriate behavior. Line 5 computes the absolute value $|I_{max}|$ of the more powerful signal perceived.

An altruistic behavior replaces the potential new task $task_n$ when the signal intensity $|I_{max}|$ is greater than $task_n$ intensity F. This comparison is adjusted by using the altruistic factor α (line 6). Then, agents may have different kinds of behavior just according to this factor.

If the altruistic behavior is selected, the new behavior is the altruism vector $\overrightarrow{\vartheta_{I_{max}}}$ computed with equation 1 section 2.2.

Vector Combination. If the selected task is a vector motion ($\overrightarrow{V_{goal}}$), it may be added with other compatible agent drives (motion-to-goal, obstacle avoidance and repulsive signals).

From perceived signals which are repulsive (set R computed line 7), the agent computes a repulsive vector using multi-agent altruism equation 2 given above in section 2.2. In practice, this vector $\overrightarrow{V_{alt}}$ is shorter than $\overrightarrow{V_{goal}}$ vector.

To avoid obstacles, a vector for sliding along obstacles is computed line 8, see Fig. 3.(a). The technique is fully presented in [14].

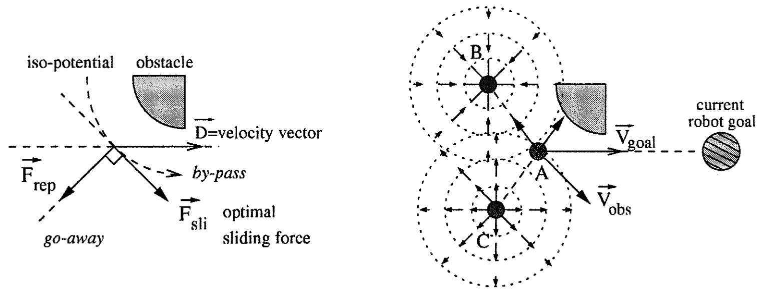

Fig. 3. (a) Sliding along an obstacle (b) Multiple influences on agent A

These three vectors are combined line 9 to compute the agent's velocity vector $\overrightarrow{V}$, where γ_1, γ_2 and γ_3 are scalar weights. The principle of combining signal and goal drives is represented in Fig. 3.(b).

Finally, in line 10, the agent performs its current task.

Note that the process of action selection and combination is computed by using only the personal satisfaction P, the perceived signals I, the altruism factor α and internal or environmental stimuli. Personal satisfaction P and interactive satisfaction I are computed from internal and external perceptions as shown below.

4 Experimental Simulations

4.1 Extractor Robots

Consider the foraging and consuming application where each robot has to leave a fixed base and explore an area which is *a priori* unknown in order to find a mine, extract raw material and transport this material back to the base. Clearly, the problem is defined by N robots (with a limited rate of extraction r), obstacles, attractive zones (mines), and a base which emits a particular signal for its localization. Moreover, each mine has a limited volume of raw material and a maximal rate of extraction. The energy available E on board each robot is bounded by E_{max}, and so periodic returns to the base power supply are necessary.

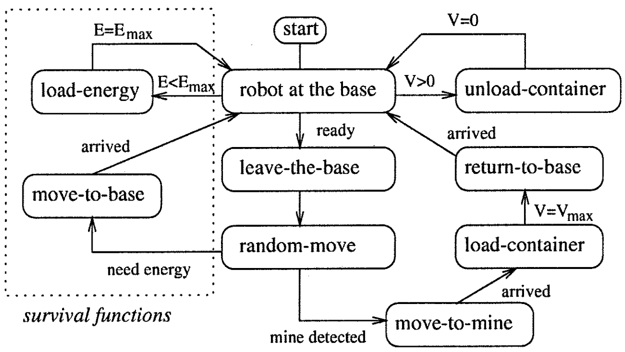

Fig. 4. Primitive behaviors of extractor robots

Extractor robots are implemented following our architecture and simple primitive behaviors (tasks). They are able to scan their environment within a short bounded distance. Figure 4 presents the global behavior of an extractor robot by giving its primitive behaviors and their stimulus. The volume of raw material extracted by a robot is noted V (with V_{max} the volume of the container).

Computation of *personal satisfaction* $P(t)$: For the moving tasks *move-to-base*, *move-to-mine* and *return-to-base*, the robot computes its progression: if it cannot move, $P(t) = -1$ else $P(t) = cos(\overrightarrow{V}_{motion}(t), \overrightarrow{V}_{goal}(t))$.

For other tasks involving moves: if the robot can move $P(t) = P(t-T)$ $+K.(0.5 - P(t-T))$ (i.e. P increases to 0.5) else P decreases to -1: $P(t) = P(t-T) - K.(1 + P(t-T))$, with $0 < K < 1$ ($K = 0.2$ for the simulations).

For extraction of raw material (*load-container*), $P(t)$ depends on the real rate of extraction $e(t)$: $P = \frac{e(t)}{r}$. With other *static* primitive actions in progress, $P(t) = 1$.

Computation of *interactive satisfaction* $I(t)$: Robots compute an increase of interactive satisfaction when they are at an exploitable mine, so that they attract other robots : $I(t) = I(t-T) + K'.(1 - I(t-T))$.

They compute a decrease of their signal when they are hindered by others robots, or when a detected mine is empty or saturated : $I(t) = I(t-T) - K'.(1 + I(t-T))$.

Otherwise, when a robot performs an altruistic behavior upon the request of an increasing signal $I_p(t)$, it also emits an increase of its interactive satisfaction: $I(t) = I_p(t)/2$. Thus, this robot may attract its idle acquaintances.

Note that while a robot has no interaction with others, its interactive satisfaction is not emitted and it tends towards 0 (neutral value of I).

4.2 Results

Cooperation, Altruism and Resolution. Situations of conflict or of cooperation are visualized as a surface with the X,Y axes reflecting the simulated environment. The Z axis indicates *the negative value* of the sum of interactive satisfaction signals emitted by agents. Peaks of the surface Z(X,Y) indicate blocking zones (because of the emission of negative values) and pits indicate attractive zones between robots. Thus, the altruistic vector is computed as if the robot was a ball rolling on this dynamic surface. Furthermore, as robot locations take a part in surface deformation, their altruistic motions tend to flatten the surface. Thus, robots avoid conflicting trajectories and move towards attractive zones (like attractive fields).

When robots have incompatible trajectories, peaks appear in the dissatisfaction surface (dashed lines 3 and 4 in Fig. 5). The robots at mines (in the left of the environment) broadcast attractive signals (pits in the surface at dashed lines 1 and 2 in Fig. 5).

In every simulation run, robots efficiently explore the environment, slide around obstacles and around other robots by using altruistic vectors. When a robot finds an exploitable mine, its interactive satisfaction increases and, as expected, neighbor robots are attracted, and by propagation they attract remote robots. If a mine is empty or saturated, the robots close to it emit a decreasing interactive satisfaction which repel new arriving robots. Finally, robots adapt their behavior to each environmental evolution: moves of robots, working out of mines and need for energy.

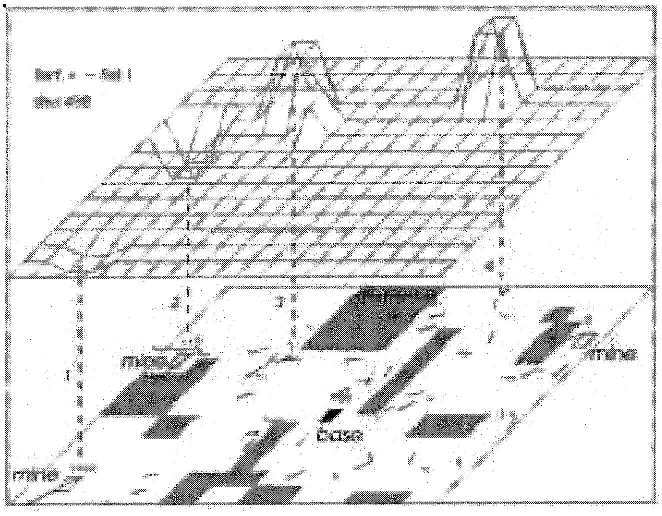

Fig. 5. Snapshot of an extractor robots simulation (50 robots, $\alpha = 0.7$).
1 robot=1 pixel, plotted with its velocity vector. Dark areas are the obstacles.

Study of Repulsion Signal-Passing. In order to evaluate performance of repulsive signals (situation Fig. 2.(b)), we have designed a long narrow environment (fig. 6). Thus, we measure the average time for at least one robot reach the mine at the right side of the environment. We have compared this time performance for different types of robots (over 500 simulations for each one).

Robots of the first type move randomly without broadcasting signals (time of reference: 226 iterations). The second type of robot avoids obstacles by using sliding vectors: 5.9% of improvement. The third type of robot includes altruism vector as goal behavior: 13.6% of improvement. In the last case, robots use all repulsive signals to compute their trajectory: 17.5% of improvement (see snapshots of these robots in figure 6).

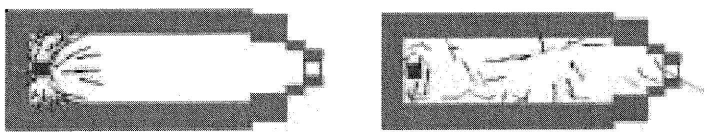

Fig. 6. Simulation of 30 altruistic robots (at iterations 12 and 170)

5 Conclusion

In this paper, we have described an architecture for multiple mobile robots. It takes into account the concept of satisfaction, and allows an agent to communicate simply its state of satisfaction and its intentions to its neighbors. The received information is directly converted into an altruistic control signal altering the otherwise selfish behavior of each agent, that drastically improves the cooperation. The architecture is simple and robust, it makes the robots immediately react to unexpected events, avoid conflicting situations and exhibit self-organization. We have demonstrated the qualities of the proposed architecture through simulations. These intensive simulations have also shown (not discussed here) that the group performance is not very sensitive to variations of the altruistic factor. Future work will include experiments with real robots and refinement of the processing of the communication signals.

References

1. R.C. Arkin. (1989) Motor Schema Based Mobile Robot Navigation, International Journal of Robotics Research, 92–112
2. R.C. Arkin. (1995) Reactive Robotic Systems, article in Handbook of Brain Theory and Neural Networks, M. Arbib, MIT Press, 8(4), 793-796
3. R.C. Arkin, T. Balch. (1997) Cooperative Multiagent Robotic Systems, AI-based Mobile Robots: Case studies of Succesful Robot systems, MIT Press
4. T. Balch, R.C. Arkin. (1994) Communication in Reactive Multi-agent Robotic Systems , Autonomous Robots, 1, Kluwer Academic Publishers, Boston, 27–52
5. R.A. Brooks. (1986) A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, 2(1), 14–23
6. A. Drogoul, J. Ferber. (1992) From Tom Thumb to the Dockers: Some Experiments with Foraging Robots, 2nd International Conference on Simulation of Adaptative behavior, Honolulu, 451–459
7. J. Ferber. (1999) Multi-Agent Systems, an introduction to Distributed Artificial Intelligence, Addison-Wesley
8. O. Khatib. (1985) Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, Proceedings of IEEE International Conference on Robotics and Automation, St. Louis 500–505
9. M.J. Mataric, M. Nilsson, K. Simsarian. (1995) Cooperative Multi-Robot Box-Pushing, Proceedings IROS '95, Pittsburgh, PA, 556-561
10. Jorg. P. Müller. (1996) The design of intelligent Agents, Lecture Notes in Artificial intelligence 1177, Springler-Verlag
11. L. Parker. (1994) ALLIANCE: An Architecture for Fault Tolerant, Cooperative Control of Heterogeneous Mobile Robots, Proceedings of International Conference on Intelligent Robots and Systems (IROS '94), Munich, 776–783
12. P. Rongier, A. Liegeois. (1999) Analysis and Prediction of the Behavior of One Class of Multiple Foraging Robots with the Help of Stochastic Petri Nets, SMC'99, IEEE International Conference on Systems, Man and Cybernetics, Tokyo Japan, 143–148
13. O. Simonin, J. Ferber. (2000) Reactive Real-Time Cooperation as a Combination of Altruism and Self Satisfaction, Proceedings of International Conference on Multi-Agent Systems (ICMAS'2000), accepted (poster), Boston
14. K. Zeghal. (1998) A Comparaison of Different Approaches based on Force Fields for Coordination among Multiple Mobiles, IEEE International Conference on Intelligent Robots and Systems (IROS '98), Victoria, B.C., Canada, 273–278