Modeling Self Satisfaction and Altruism to handle Action Selection and Reactive Cooperation

Olivier Simonin and Jacques Ferber
LIRMM - Université Montpellier II,
161 rue Ada, 34392 Montpellier Cedex 5, France
{simonin,ferber}@lirmm.fr

Abstract

This paper proposes a method for integrating cooperative behaviors among distributed autonomous agents based on local communications. We define agent satisfaction as a signal to handle action selection and cooperative interaction. The resolution of spatial problems is handled with an altruistic behavior defined with the vector field model. This solution ensure the spatial coordination of multiple agents in a dynamic system and the selection of an altruistic behavior when necessary. After presenting the agent architecture we describe experimental results. They illustrate the ability of the model to achieve efficient navigation, adaptive and reactive cooperation in collective robotics application.

1. Introduction

The reactive approach to autonomous robot design that has been developed since about fifteen years has its roots in animals models (see (Arkin, 1995)). Indeed, insect societies and many animals show great capabilities to perform cooperative tasks, to adapt and survive in uncertain environments and to have reactive behaviors. For instance, Ant colonies are composed of simple creatures which can perform tasks by swarm intelligence (Deneubourg et al., 1991) (Drogoul and Ferber, 1992).

Many animats which have been implemented with *reactive behavioral control* technique (Arkin, 1995) have only *local* interactions with the environment and the others agents. But they can perform collective tasks as navigation (Arkin, 1989) , foraging (Drogoul and Ferber, 1992) (Balch and Arkin, 1994) and distributed problem solving (Steels, 1989) (Drogoul et al., 1991) (Ferber, 1995b). These reactive architectures are able to provide *self-organization, reactivity and adaptability.*

However, the poor level of communication and cognition of reactive architectures cannot easily handle goal driven behaviors (Muller, 1996) and not allow explicit cooperation to perform complex tasks. On the contrary, deliberative and hybrid architectures use planning and protocols of communication to perform complex or cooperative tasks. But they lose any reactivity and robustness abilities (due to communication failures, time for planning, imperfect sensors, etc...).

The aim of this work is to provide a robust intentional cooperation into a reactive architecture without lose its intrinsic qualities. We do not use learning techniques but an efficient communicating model. This research is carried out in order to develop small and large scale swarm robotic systems.

1.1 Signals and Communication

Communication between agents may be *explicit* or *implicit* (also called indirect communication (Mataric, 1995)).

For instance, implicit communication is used in insect societies. Insects communicate through the environment by laying landmarks and pheromones. But this technique is difficult to implement in real robotics and not allow fast communication.

Explicit communication requires emission of signals/messages or broadcast of information (also called direct communication (Mataric, 1995)). But certain types of cooperative behavior need directed communication (one-to-one or one-to-many) which are implemented in deliberative/hybrid architectures. However, these goal communications are costly in time and poorly robust, because information must be encoded, transmitted, received and coded (Arkin, 1995).

Nevertheless, many animals succeed to doing complex tasks without using a high level of communication but just by emitting and reading signals. As it is emphasized in (McFarland, 1987) signals used by animals are closely link to their social organization. For example, birds emit different type of songs to send an alarm signal, a localization, a recognition, etc... Animal signal spreading have interesting characteristics such as spatial limited communication, intensity meaning, redundant information (robustness) and reactivity.

We investigate this type of communication to improve reactive architecture in order to handle robust communication and reactive cooperation.

1.2 Self Interest and Altruism

Animals use communication to influence the behavior of members of their species (McFarland, 1987). When an information is emitted, the sender expects a specific reaction from the receiver. This reaction may be an action which satisfy a waiting of the sender or just a reaction without reward for the sender (an altruistic behavior). For instance, an alarm signal is an altruistic behavior, whereas a courtship display is made for an expected behavior from the receiver. Note that in these two cases, both of these behaviors are made for self interest and altruism because they ensure collective survival and as a consequence the individual survival (McFarland, 1987).

The behavior of a creature is driven by motivations such as survive, feeding, hunting, reproducing. At any time, it must choose an appropriate behavior to *satisfy* one or many of these needs.

The underlying idea of our architecture is to use the notion of *satisfaction* (Ferber, 1995b) to perform action selection. This satisfaction is defined as the combination of the agent *self satisfaction* (i.e. its self interest), with a so-called *empathy satisfaction* which takes into account the satisfaction of other agents.

By taking into account the satisfaction of its acquaintances, an agent can have an *altruistic behavior* thus allowing for intentional cooperation between agents.

This paper introduces an agent architecture which handles cooperation between agents at the reactive level, without requiring any deliberative abilities but with emission of a simple signal.

This model addresses the following abilities:

- autonomous navigation and real-time reaction to spatial conflicts,
- selection and combination of several compatible actions,
- real-time and decentralized cooperation (i.e. spatial coordination and altruistic behaviors),
- robust and simple architectural design.

The paper is organized as follows: We define in Section 2 the concept of agent satisfaction and in Section 3 the altruistic behavior computed into the vector model. Section 4 describes the reactive satisfaction based architecture. We then provide in Section 5 experimental results and a discussion on the quality of the model. We close in Section 6 with finals remarks and some directions for future work.

2. Agent Satisfaction

2.1 Cooperation and Satisfaction

Cooperation in multi-agent systems can be seen as a system in which cooperative methods lead to an increase of the group performance and thus the probability for an individual to maximize its self objectives (Ferber, 1995a). Moreover, as L. Parker emphasizes in (Parker, 1994), many robotic applications are inherently distributed in *space*, time, or functionality, thus requiring a distributed solution.

From the designer's point of view, the animat behavior must satisfy *goal-oriented*, *conservative* and *cooperative* functions. We abstract the required behavior of a cooperative robot as follows: *the robot must satisfy its individual goals while minimizing negative interactions (conflicts) and maximizing positive interactions (cooperation) with other agents and the environment.*

The principle of our agent model relies on the maximization of agent satisfactions. At any time, the agent try to maximize :

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

As *motions* are involved in a majority of situated agent tasks, we investigate in priority spatial cooperation and spatial conflicts resolution. We define the agent behavior as a *combination of selected goals* which can either be *individual goals* or *altruistic responses to requests for cooperation.*

2.2 Sub-satisfactions

In our model, *agent satisfaction* is composed of three different sub-satisfactions: ***personal satisfaction***, ***empathy satisfaction*** **and** ***interactive satisfaction***.

- the ***personal satisfaction*** P measures only the *progress* of the agent task (from its sensors).
- the ***empathy satisfaction*** E is the average value of personal satisfaction of its acquaintances, it expresses the *altruism* of the agent.
- the ***interactive satisfaction*** I results from agent interactions. An agent computes the interactive satisfaction as a reaction or an intention to other agents.

Note that personal and interactive satisfactions depends on the progress of agent individual goal (they define the self satisfaction). For instance, let us consider an agent which tries to push an object. When the object moves,

	Agent A	Agent B	Situation	Type of interaction	Altruist behavior of B
1	$I_A \searrow$	$I_B \rightarrow$	A hinders B	conflict	avoid A
2	$I_A \searrow$	$I_B \searrow$	mutual hindrance	conflict	mutual avoiding
3	$I_A \nearrow$	$I_B \searrow$	incompatible interests	conflict	move towards A
4	$I_A \nearrow$	$I_B \rightarrow$	B needs A	cooperation	move towards A
5	$I_A \nearrow$	$I_B \nearrow$	mutual needs	cooperation	move towards A
6	$I_A \rightarrow$	$I_B \rightarrow$	indifference	neutral	continue action

Table 1: Interaction between two agents

its personal satisfaction increases. Its interactive satisfaction increases when it needs help to move the object, or decreases if an agent hinders its progression. Its empathy satisfaction is positive if its acquaintances are satisfied too.

Formally, we set the instantaneous satisfaction of an agent i at time t:

$$Sat_i(t) = (1-\alpha).P_i(t) + \alpha.E_i(t) \quad (1)$$

α is the *altruistic factor* of the agent ($\alpha \in [0,1]$) and P_i and E_i are defined on the interval $[-1,1]$.

To handle cooperation, agents compute *their interactive satisfaction* I. This value is computed as an intention in order to alter agent interactions. As presented below, only P *and* I sub-satisfactions must be computed in our architecture to handle selection, combination and cooperation of actions.

3. Modeling agent altruism

Agents broadcast only their interactive satisfaction I because it expresses an intention or a reaction to other agents behaviors. 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.

We call $I_i(t)$ the level of interactive satisfaction emitted by the agent i at time t. Let $\Delta I_i(t)_T$ be the function of the interactive satisfaction variation:

$$\Delta I_i(t)_T = I_i(t) - I_i(t-T) \quad (2)$$

Each agent which receives a signal I can compute this variation. It represents the satisfaction evolution of an agent relative to its acquaintance actions between times $t-T$ and t. The sign of ΔI gives 3 types of signal evolution : positive $\nearrow$, negative $\searrow$ or constant $\rightarrow$.

Note that these signals are a type of state communication (robots are able to detect an internal state of others). It is a more robust means than goal communication. As in ALLIANCE model (Parker, 1994), we use a simple form of broadcast communication to allow agents to inform others of their current activities. But, in our satisfaction model, we can *continually broadcast* information, without having to encode, transmit and decode a message.

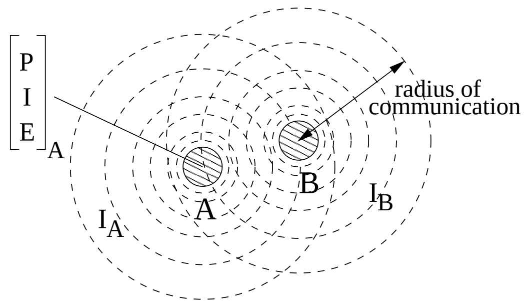

Figure 1: broadcasting of agent signals

3.1 Inter-agent altruism

As we emphasize above, agents motions are embedded in a lot of tasks. In table 1 the different types of spatial interactions between two agents are given for various evolution of their respective interactive satisfaction. The first column contains I evolution computed by A relatively to agent B presence. The symmetrical case (B reaction to A presence) is given in the second column. As the satisfaction evolution can take 3 "directions", the number of possible situations is 9, but to avoid symmetrical situations only 6 cases are tabulated.

Each situation is described within the third and the fourth column. The last column gives the required behavior of B in order to increase the satisfaction of A (or to decrease the dissatisfaction of A). This behavior clearly depends on the other agent interactive satisfaction variation (ΔI_A).

In the vector model, this cooperative behavior is defined with the *altruism vector* $\overrightarrow{\vartheta}(t)$. Formally, the altruism vector of an agent (B) relative to another agent (A) $\overrightarrow{\vartheta_{B/A}}(t)$ is computed as follows:

$$\overrightarrow{\vartheta_{B/A}}(t) = k.S(\Delta I_A(t)_T).\frac{|\, I_A(t)\,|}{\left\|\overrightarrow{AB}\right\|^2}.\overrightarrow{BA} \quad (3)$$

$$S(x(t)) = \begin{array}{ll} Sign(x(t)) & \text{if } x(t) \neq 0 \\ Sign(x(t-T)) & \text{if } x(t) = 0 \end{array}$$

Equation 3 is like a force deriving from a repulsive field. Thus, the reaction becomes higher when the agent comes close to an obstacle or a goal. Our approach consist to transform the signal of interactive satisfaction

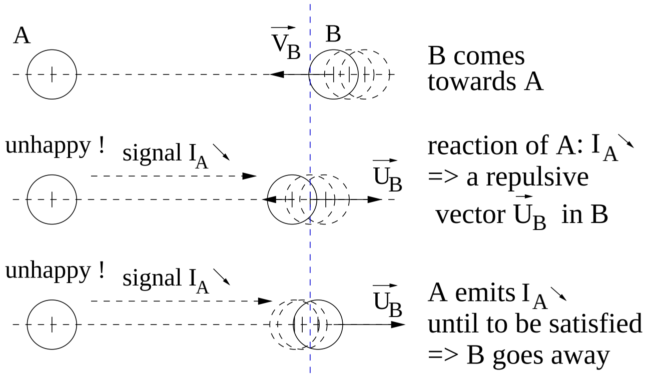

Figure 2: repulsive reaction of A relatively to B coming

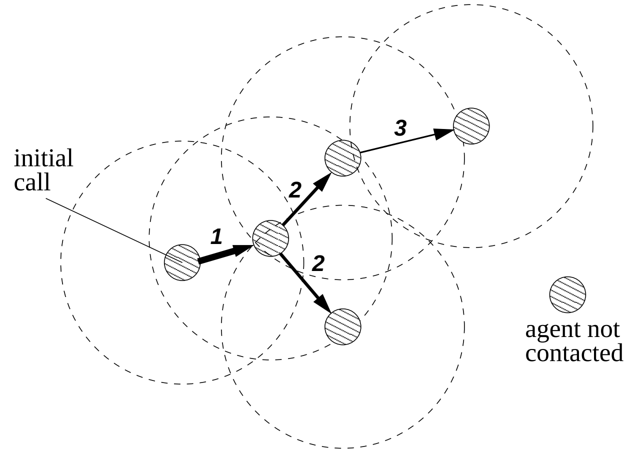

Figure 3: Propagation of an attractive signal from an agent to its neighbors (numbered in emission order)

into a mathematical vector directly added to the agent motion vector. Because of the reactive communication, agents move in real time into dynamical vector fields (see section 5.3.2).

The interest of this equation 3 lies in the fact that it can be applied in a variety of situations. The vector $\overrightarrow{\vartheta_{B/A}}(t)$ is obtained when

- just one agent moves (see figure 2 as an example)
- both agents move
- none of the agents move but they perceive interactive satisfactions not null.

Because ΔI_A may be null, we have defined a sign function S to avoid a null vector $\overrightarrow{\vartheta_{B/A}}(t)$. This vector must be computed even when ΔI_A is null to obtain the continuation of a reaction. If ΔI_A is null (stationary satisfaction) the equation 3 must be computed with the last non null sign of ΔI_A. Indeed, the last reaction remains the most significant of current situation.

To complete the validation of the equation 3 we examine situations where both agents are mutually influenced (see table 1 rows 2, 3 and 5):

row 2 $Sign(\Delta I_A)$=$-$ and $Sign(\Delta I_B)$=$-$, we obtain two opposite repulsive vectors. Then agents go away from each other and so increase their satisfaction level.

row 5 $Sign(\Delta I_A)$=+ and $Sign(\Delta I_B)$=+, we obtain two convergent attractive vectors. Agents are mutually attracted and then their satisfaction increases.

row 3 $Sign(\Delta I_A)$=$-$ and $Sign(\Delta I_B)$=+ (or inversely +/-), we obtain an attractive vector and a repulsive vector with the same direction ! It is the case of incompatible interests. Then, we can define different solutions as amplify the attractive vector to avoid a chase behavior.

Propagation from agent to agent

Propagation in repulsion situations : 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 may move.* This implicit propagation of dissatisfaction to all involved agents release the deadlock situation.

We can also use *propagation for attractions between agents:* When an agent perceives a signal with a positive increase, it may decide to follow the call. In this case, the agent emit an attractive signal in order to call its own acquaintances. The force of the signal variation must decline during the propagation in order to limit the number of attracted agents. But this number does not depend on the distance between the initial sender and other agents. It just depends on a trigger threshold (an intensity of the signal) and inter-agent distances. This distance must be smaller than the radius of communication to propagate information from agent to agent. See figure 3 for a schematic representation.

3.2 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 divide the multi-agent problem in a set of inter-agent problems. A first equation generalizing the equation 3 may be written (altruistic reaction of agent B close to N acquaintances called Λ):

$$\overrightarrow{\vartheta_{B/\Lambda}}(t) = \sum_{j\in\Lambda} \overrightarrow{\vartheta_{B/j}}(t) \tag{4}$$

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

However, application of this equation can reach to different quality of results following the way of communication.

In the presented model, an agent emits just a simple signal (its value of interactive satisfaction). Thus it can not share several intentions to reply to several simul-

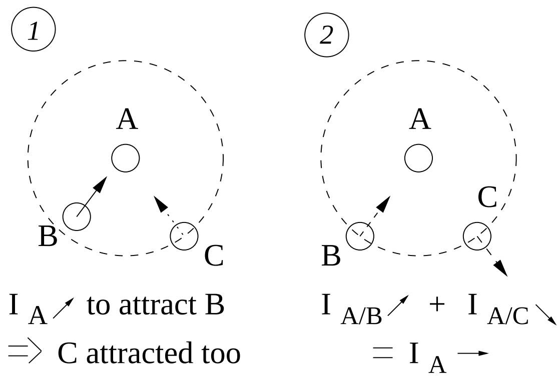

Figure 4: problematical situations

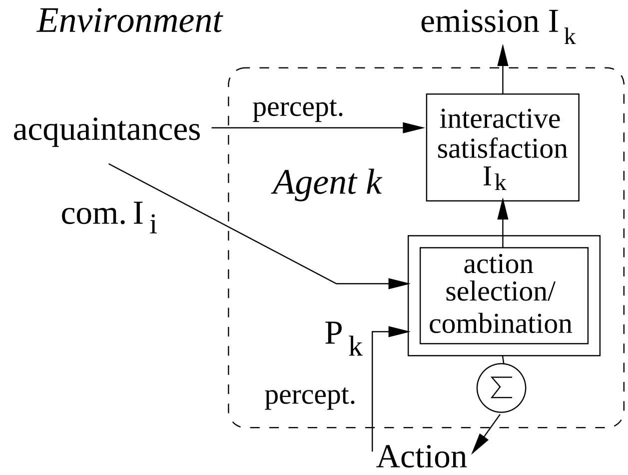

Figure 5: Principle of the architecture

taneous influences. Before studying a better solution, shortcomings of the actual solution are identified:

- With homogeneous agents: As agents have the same abilities, any agent can help or hinder another. Then, a signal reaching multiple agents is not really a problem. For instance, consider a robot which needs help from another to perform its task. To be helped it emits a call signal to its acquaintances (by increasing its interactive satisfaction). If several robots are simultaneously attracted, the first arriving will help the calling agent. Others attracted robots will be free when the signal will be stopped.

- With heterogeneous agents: An agent can need help from specific acquaintances but not from all. We present conflict situations due to the simple signal method within different diagrams. Figure 4 case 1 illustrates that a signal of A ($I_A \nearrow$) in order to attract B may attract C too. In the second case of fig. 4, two opposite influences on A give by addition a null variation of I_A (which precludes any reaction of B and C). However, this case can be handled by allowing a priority to the attractive signal (it is often more important to attract an expected agent than to push away hindering agents)

These ambiguous situations may occur because communication are performed with a single and simple signal. However, this method is intentionally used to avoid fragile and costly solutions (in memory and time). We present a reactive solution to manage this problem with limited changes in the previous solution.

3.2.1 The curvature of the satisfaction signal

This solution allows agents to perform a more fine reading of the satisfaction signal. The way of communication is always a broadcasted simple wave.

The problem for an agent, while "hearing" the variation of interactive satisfaction from another is to identify the intention addressed to it. In fact, an altruistic agent tries to "understand" what the others expect from it: go to him or go away ? To improve the agent understanding, we compute the sign of *I curvature* (i.e. gradient of ΔI) instead of the sign of ΔI. In consequence, agents must perceive two times the interactive satisfaction to compute the gradient of ΔI. When an agent is already subject to several influences from other agents, the sign of the curvature expresses the impact of a new coming agent. Let ΩI be the gradient of the interactive satisfaction variation:

$$\Omega I(t)_T = \Delta I(t)_T - \Delta I(t-T)_T$$
and if $\Omega I(t)_T = 0$ then $\Omega I(t)_T = \Delta I(t)_T$

Thus, the general equation is $\overrightarrow{\vartheta_{B/\Lambda}} = \sum_{j \in \Lambda} \overrightarrow{\vartheta_{B/j}}$
$= \sum_{j \in \Lambda} k.S(\Omega I_j(t)_T).\frac{|I_j(t)|}{\left\|\overrightarrow{BJ}\right\|^2}.\overrightarrow{BJ}$

Generally, several agents do not come synchronously into the field of perception of another agent. Thus, each agent which computes the curvature of the received satisfaction can easily estimate the impact caused by its own presence.

4. Principle of the architecture

Our architecture is based on a classical schema-based reactive control (Balch and Arkin, 1994) *using vector combination* (Arkin, 1989)(Zeghal, 1998). Actions are selected and computed from proprioceptive information (energy, conservative functions) and exteroceptive information (obstacles, goals), with the addition of a new vector called *altruistic vector*. It represents the cooperative part of the agent behavior.

4.1 Task Selection

Each task (primitive behavior) is an *action* which may be triggered by an internal or environmental *stimulus* (like into (Maes, 1991) (Drogoul and Ferber, 1992) architec-

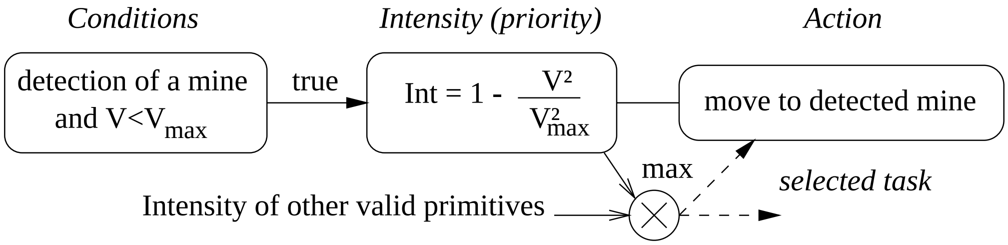

Figure 6: Primitive *Move-to-mine* and the principle of primitive selection

tures). The test to release an action is defined by a set of boolean conditions on perceptions, noted $Cond(task)$. Moreover, each condition induces a measure of intensity computed from perceptions and called $Int(task)$ (defined on the interval $[0,1]$). For instance, figure 6 gives the representation of the *move-to-mine* primitive.

Some tasks are functions for *survival*, i.e. they are useful for robot working. Thus, these tasks *have priority on others* (by a subsumption control (Brooks, 1986)) .

If no survival function is released, the agent must continue its current task ($task_c$) or select a new one.

We describe now the computation of the new candidate task (called $task_n$).

First, we select the task which have the maximum perceived intensity among triggered tasks. Formally, this task, called $task_m$, must fit
$\forall_{i \neq m} Int(task_m) \geq Int(task_i)\, and\, Cond(task_m) = true$

If $task_m$ has an intensity greater than the current task performance index (= personal satisfaction P) it becomes the new candidate task. Formally,

$$task_n = \begin{cases} task_m & if \quad Int(task_m) > P \\ task_c & else \end{cases}$$

However, this new candidate task may be replaced by an altruistic behavior. For this purpose, we compute the intensity F of $task_n$ selection.

$$F = \begin{cases} Int(task_n) & if \quad task_n = task_m \\ P & else \end{cases}$$

4.2 The Altruistic behavior

As agents perceive interactive satisfactions I from others, they can reply to those signals by exhibiting an appropriate behavior. Let $|I_{max}|$ be the absolute value of the more powerful signal perceived. An altruistic behavior may replace the potential new task if

(1) an altruistic action is already performed

or (2) $|I_{max}|$ is greater than $task_n$ intensity F.

In the first case (1), the agent keeps its current altruistic behavior (ensuring continuity in agent actions). However, the relative perceived signal must keep the same sign variation, else the agent compute the second possibility (2).

In the second case, $|I_{max}|$ is compared with F to select the new current task $task_c$:

$$task_c = \begin{cases} \overrightarrow{\vartheta_{I_{max}}} & if \quad \alpha.|I_{max}| \geq (1-\alpha).F \\ task_n & else \end{cases}$$

where $\overrightarrow{\vartheta_{I_{max}}}$ is computed with equation 3 section 3.1. The comparison is adjusted by using the altruistic factor α. Then, agents may have different kind of behavior just according to this factor.

4.3 Vector Combination

Let us consider an agent p which carry out a task involving motions in the environment. If the agent p perceives a signal I_k from an agent k, it can compute an altruistic vector $\overrightarrow{\vartheta_{p/k}}$. This vector may replace the current task or may be just added with other compatible drives to improve the traffic.

We consider that only goal motion, obstacle slidings and repulsive signals may be combined. Formally, the agent's velocity vector $\overrightarrow{U_p}$ is computed as follows :

$$\overrightarrow{U_p} = \gamma_1.\overrightarrow{V_{goal}} + \gamma_2.\overrightarrow{V_{obs}} + \gamma_3.\overrightarrow{\vartheta}_{p/R(I_1,I_2,..)}$$

where γ_1, γ_2 and γ_3 are scalar weights, $\overrightarrow{V_{obs}}$ is the vector for *sliding* along static obstacles, $\overrightarrow{V_{goal}}$ is the current goal vector, and $\overrightarrow{\vartheta}_{p/R(I_1,I_2,..)}$ is an altruistic vector computed from perceived repulsive signals.

The computation to avoid an obstacle is performed by the 90∘ rotation of the repulsive vector. The sliding technique is fully presented in (Zeghal, 1998).

As shown above $\overrightarrow{V_{goal}}$ is either an altruistic behavior $\overrightarrow{\vartheta_{I_{max}}}$ or the vector computed according to $task_n$ goal.

In order to compute $\overrightarrow{\vartheta}_p$ let $\{I_1, I_2, ..\}$ be the set of signals perceived by the agent p. Then, let R be the set of *repulsive* signals : $R = \bigcup\{I_i \,/\, \Delta I_i < 0\}$.

For each signal which is repulsive, agent p computes a sliding altruistic vector in order to improve its acquaintances motion. Thus $\overrightarrow{\vartheta}_{p/R(I_1,I_2,..)}$ is computed like the multi-agent altruism vector defined above in section 3.2 with equation 4: $\overrightarrow{\vartheta_{p/R}} = \sum_{j \in R} \overrightarrow{\vartheta_{B/j}}$.

Note that the process of action selection/combination is computed by using only the *personal satisfaction* P,

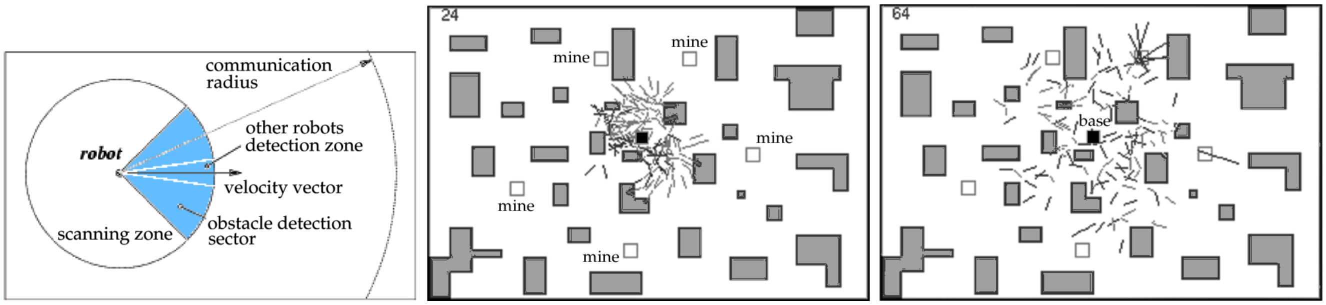

Figure 7: (a) robot perceptions (b) and (c) simu. in progress with 100 robots, 5 mines and 18% obstacles

the *perceived interactive satisfactions* I, the altruism factor α and internal or environmental stimuli (see fig. 5).

Personal satisfaction P and interactive satisfaction I are computed from internal and external perceptions. In the next section, they are expressed in details within a concrete application.

5. Experimental results

5.1 *Foraging and Consuming*

As an illustrative example of cooperation and coordination a collection of robots must accomplish, consider homogeneous extractor robots. Each robot has to leave a fixed base and explore a space which is *a priory* unknown in order to find a mine, extract raw material and carry this material to the base. Clearly, the problem is defined by

- N agents situated in a 2D limited area, which can extract material from mines (with a limited rate r),
- Obstacle zones,
- Attractive zones = mines
- A base which emits a particular signal for its localization.
- Each mine has a limited volume of raw material and a maximal rate of extraction (giving a dynamic problem of optimal extraction). Moreover, energy available on board each robot is bounded, and so periodic returns to the base power supply are necessary.

These terms involve several type of problems: coordination of mobile robots in a constrained environment, handling robot's energy and optimizing the number of robots at mines.

5.2 *Extractor Robots*

Extractor robots are implemented following our architecture and simple primitive behaviors (tasks). They use the first type of communication presented in section 3. The volume of raw material extrated by a robot is noted V (with V_{max} the volume of the container).

As an example of simple robot task, the primitive *move-to-mine* is presented in fig. 6 with its stimulus conditions, its intensity function and its potential action. Robot's behavior includes other primitives: *leave-the-base, move-to-mine, load-container, carry-the-material, unload-container, go-to-base, load-energy and move-randomly.*

As presented in the architecture, each motion behavior is combined with obstacle avoidance and altruistic vectors (section 4.). Robots are able to scan their environment within a short bounded distance. Obstacle and agent detections are performed whitin sectors as shown in figure 7.(a) .

Computation of *personal satisfaction* $P(t)$:

For the *go-to-base* primitive, the robot compute its progression : if it cannot move $P(t) = -1$ else $P(t) = cos(\overrightarrow{V_{motion}}(t), \overrightarrow{V_{base}}(t))$. The same technique is used for the tasks *move-to-mine* and *carry-the-material.*

For others 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-time rate of extraction ($e(t)$): $P = \frac{e(t)}{r}$.

With other *static* primitives actions in progress, $P(t) = 1$.

Computation of *interactive satisfaction* $I(t)$:

When robots are at an exploitable mine, they must call other agents. Thus, they compute an increase of interactive satisfaction: $I(t) = I(t-T) + K' * (1 - I(t-T))$.

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. Formally, if $I_p(t) > threshold$ and $\Delta I_p(t) > 0$, $I(t) = I_p(t)/2$. Thus, this robot may attract its idle acquaintances.

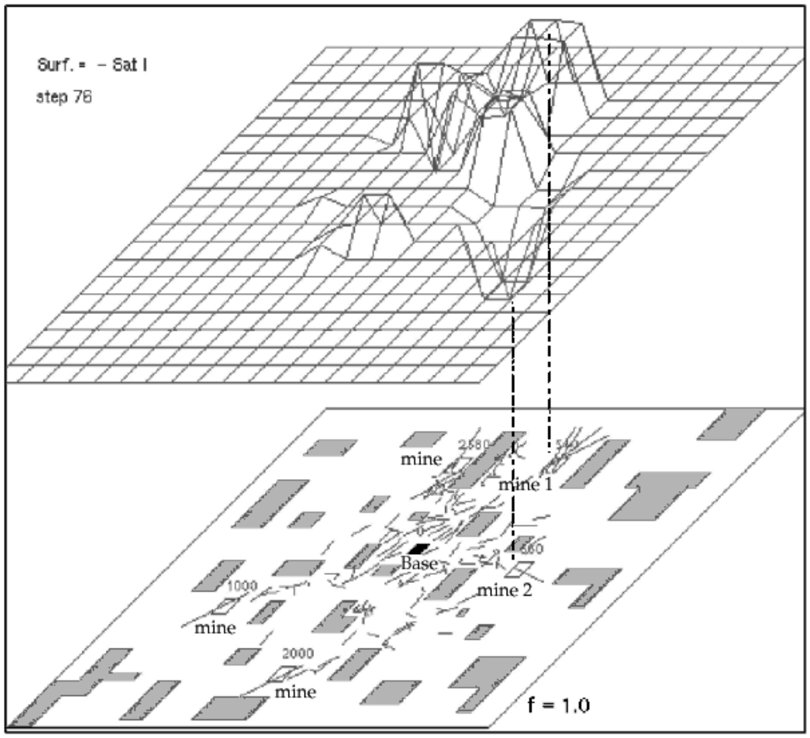

Figure 8: Snapshot of a simulation (with 100 robots, 5 mines, 18% of obstacles) with its dissatisfaction surface.

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). Otherwise, if this normalization was emitted, other robots could understand it as an intentional signal variation.

5.3 Simulations and Quality of the Model

5.3.1 Simulation Environment

The robots are simulated using the Madkit Platform (Ferber and Gutknecht, 1998). Each step is one iteration of the program that calculates the robots' next position. The environment is a rectangular area in which obstacles are composed of dark rectangles. The base is a black filled square and mines are empty squares. Each Robot is simply defined by a pixel (unit) and represented with its different vectors (motion direction and obstacle sliding). The line of the motion vector is doubled when the robot detect/exploit a mine. Robots have a scanning radius of 10 units, a communication radius of 30 units, and an altruistic factor α equal to 0.7.

5.3.2 Resolution of the problem

We first briefly describe results obtained with different instances of the studied problem. We have run simulations with different configurations. The parameters are: obstacles density (5%, 20% and 40%), number of mines (2 to 10) and number of robots (10, 50, 100).

In every simulation run, robots efficiently explore the environment and slide around obstacles. They frequently compute an altruistic behavior to avoid mutual hindrances, and as a consequence they are always *homogeneously distributed* (as shown in fig. 7.(c)).

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 (because of the limited rate), the robots close to it emit a decreasing interactive satisfaction which repel new arriving robots.

Robots explore, extract and carry the raw material until mines are worked out. Moreover, propagation of information from agent to agent improves time performance in comparaison with the same robots *without communication*. With the problem configuration of figure 8 we have obtained a gain in time performance of 20%. But, this gain is sensitive to many environmental parameters: raw volume, mines rate, number of robots, radius of communication, obstacles, etc...

When robots need energy, the survival function *go-to-base* is triggered and they come back to the base. Finally, robots adapt their behavior to each environment evolution : moves of robots, working out of mines and need for energy.

The social behavior of these robots exhibits the property of reactive and self-organized systems: distributed problem solving (efficient spatial distribution, adaptative task allocation), reactivity and robustness.

Cooperation and Altruism

Situations of conflict or of cooperation are visualized as a surface with the X,Y axis reflecting the simulated environment. The Z axis indicates *the negative value* of the sum of interactive satisfaction signals emitted by agents (= surface of dissatisfaction Z(X,Y)).

Peaks of the surface indicate hindrances between robots or deadlock zones (because of the emission of negative values) and pits indicate attractive zones (call signals). Thus, the altruistic vector is computed as if the robot was a ball rolling on this dynamic surface, i.e. rolling down from peaks to pits. Furthermore, as robot locations take a part in surface deformation, their altruistic motions tends to flatten the surface. Thus, robots avoid conflicting trajectories and move towards attractive zones (like attractive fields).

The snapshots presented in figure 8 illustrates attractions and repulsions emitted by robots. The mine numbered 1 at the top right side of the environment is saturated, then a peak appears in the dissatisfaction surface. On the other hand, the robot close to the mine 2 (at the right side of the environment) emit a call signal which create a pit in the surface.

Fluidity of the traffic

As one goal of this work is to perform reactive spatial coordination of mobile robots, the traffic performance must be measured. This performance is defined as the average fluidity of agents movements at each step of the simulation. We measure successful motions over the number of trials:

$$fl(t) = \frac{\sum sucessful\,motions\,(t)}{\sum trials\,(t)}$$

The curve $y = fl(t)$ allows us to study the evolution of the traffic into simulated systems. So, we can detect deadlocks when they happen. With many robots (over 20), we have generally observed sharp decrease in the beginning of simulations (due to hindrances between them).

Figure 9 presents curves of the fluidity computed for three different type of agent architecture with the problem configuration of fig. 8. Simulations have been computed with the same initial state (100 robots) and during the first 100 iterations. The curve noted 1 has been computed with satisfaction-based robots. The second curve results from robots which use obstacle sliding navigation but whitout emitting signals. The last curve, noted 3, results from simple robots using only obstacle repulsion to navigate (Khatib, 1985).

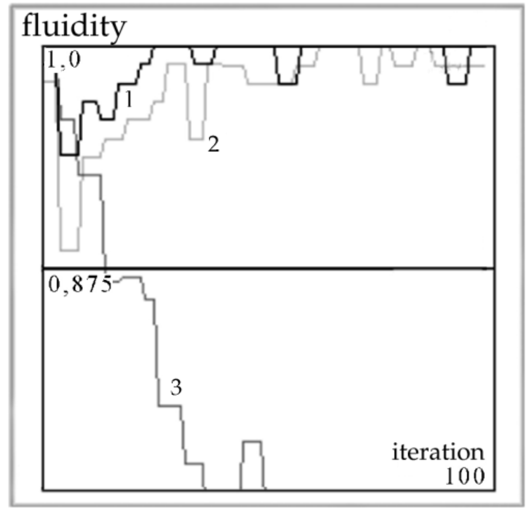

Figure 9: Curves of fluidity for (1) satisfaction-based robots, (2) sliding robots, (3) repulsive robots (problem conf. fig. 8)

First, it is clear that sliding obstacles (and robots) is better than use only repulsive vectors into the trajectory computation (the traffic fluidity decreases under 0.75 when robots not use the sliding technique, see curve 3 fig. 9). Secondly, satisfaction-based robots perform a fluid navigation ($fl(t)$ is close to 1, see curve 1). In other words, altruistic behaviors are easily integrate to the computation of robot trajectory and they improve a bit the fluidity.

General evaluation

The model is evaluated considering performance metrics proposed by Balch and Arkin in (Balch and Arkin, 1994):

- Cost : The architecture is simple and robust (vector combination and communication of simple redundant signals), then not expensive. Moreover, the cooperative and adaptive abilities minimize the number of robots required to perform robotic applications.
- Time : In (Balch and Arkin, 1994), this metric is evaluated as the maximum number of robots that can operate without interference. Our model, which is cooperative and navigation oriented provides high-performances for spatial interactions (see fluidity factor results).
- Energy : Implementation of our architecture shows that the model handles easily the problem of energy recharge. Moreover, as robots are cooperative and try to avoid spatial conflicts , *they avoid wasting energy situations*, i.e. they minimize the amount of energy used.

- Reliability/Survivability : As the model use a *reactive behavioral control* for action selection, each robots can use its working primitives even if some of them fail. The emission of simple redundant signals guarantee a robust communication, and so a good survavility of robots into hostile environments.

6. Conclusion and perspectives

The model presented in this paper shows the feasibility of using agents intentions in a reactive architecture to solve the problem of action selection/combination. We have seen that reactive cooperation relies on simple signals and the propagation of satisfactions between agents.

This model remains within the vector field approach and provides for real-time cooperation. Thus, cooperative behavior is not computed at a deliberative level but within the reactive level. We have shown that we can combine conservative functions, spatial coordination and altruistic behavior within the same framework.

Experimental results show that the model is efficient and have adaptive capabilities. Satisfaction based robots easily adapt to the current actions of other agents and to a dynamically changing environment. Moreover, the model ensure a high level of traffic fluidity, propagate relevant information and may remove a spatial deadlock in a very short time.

Future works will address the study of parameter variations effect on performances, the introduction of learning and its application to real mobile robots.

References

Arkin, R. (1989). Motor schema based mobile robot navigation. *International Journal of Robotics Research*, pages 92–112.

Arkin, R. (1995). Reactive robotic systems, article in handbook of brain theory and neural networks. 8(4):793–796.

Balch, T. and Arkin, R. (1994). Communication in reactive multi-agent robotic systems. *Autonomous Robots*, 1:27–52.

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.

Deneubourg, J., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1991). The dynamics of collective sorting robot-like ants and ant-like robots. *Proceedings of the first international conference on simulation of adaptative behavior*, page 356.

Drogoul, A. and Ferber, J. (1992). From tom thumb to the dockers: Some experiments with foraging robots. In *2nd Int. Conf. On Simulation of Adaptative behavior*, pages 451–459, Honolulu.

Drogoul, A., Ferber, J., and Jacopin, E. (1991). Pengi : Applying eco-problem solving for behavior modeling in an abstract eco-system. In Mosekilde, E., (Ed.), *European Simulation Multiconference.*

Ferber, J. (1995a). Basis of cooperation in multi-agent systems. In *Proc. of the 95 European Conference on Cognitive Science, Saint Malo.* INRIA.

Ferber, J. (1995b). *Les Systemes Multi-Agents, Vers une intelligence collective, InterEditions, 1995 or Multi-Agent Systems, an introduction to Distributed Artificial Intelligence, Addison-Wesley, 1999.*

Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agent systems. In *Proceedings of the 3rd International Conference on Multi-Agent Systems*, pages 128–135, Paris.

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 500–505, St. Louis.

Maes, P. (1991). A bottom-up mechanism for behavior selection in an artificial creature. *Proceedings of the first international conference on simulation of adaptative behavior*, page 238.

Mataric, M. (1995). Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16(2-4):321–331.

McFarland, D. (1987). *The Oxford Companion to Animal Behaviour.* Oxford University Press.

Muller, J. P. (1996). The design of intelligent agents. *Lecture Notes in Artificial intelligence*, 1177.

Parker, L. (1994). Alliance: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS '94)*, pages 776–783, Munich, Germany.

Steels, L. (1989). Cooperation between distributed agents trough self-organization. *Journal on robotics and autonomous systems.*

Zeghal, K. (October 13-17 1998). A comparaison of different approaches based on force fields for coordination among multiple mobiles. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Victoria, B.C., Canada.