Adaptive communication promotes sub-system formation in a multi agent system with limited resources

Di Prodi Paolo
University of Glasgow
Dep. of E&E Engineering,
Glasgow, G12 8LT, UK
epokh@elec.gla.ac.uk

Bernd Porr
University of Glasgow
Dep. of E&E Engineering,
Glasgow, G12 8LT, UK
b.porr@elec.gla.ac.uk

Florentin Wörgötter
University of Göttingen
Bernstein Center of Comp. Neuro.
Germany
worgott@bccn-goettingen.de

Abstract

In society subsystems are formed to reduce uncertainty. Subsystems are composed by agents with a reduced behavioural complexity. For example in society there are people who produce goods and other who distribute them. In this paper we show that sub-systems emerge when the agents are able to learn and have the ability to communicate. Both the behaviour and communication is learned by the agent and is not imposed on the agent. Here the task is to collect food, keep it and eat it until sated. Every agent communicates its satedness state to neighbouring agents. This results in two subsystems whereas agents in the first collect food and in the latter steal food from others. The ratio between the number of agents that belongs to the first system and to the second system, depends on the number of food resources which are limited in space and time.

1 Introduction

Agents face different uncertainties in their environment which have to be dealt with. The simplest solution is an appropriate reflex which guides the agent from or to a certain object, for example a wall or food, respectively [1]. Learning enables the agent to anticipate reflexes and to generate anticipatory behaviour [14]. This, however, poses a problem because when all agents learn, they change their behaviour all the time which renders them more and more unpredictable [5]. Luhmann proposed that the creation of subsystems will overcome this problem. Within these subsystems, agents perform more predictably, by reducing their behavioural complexity. In this work we show that subsystems are formed only when agents are able to learn to communicate. Communication is learned at the receiver side. Paper is structured in: description of the agents, world and signals involved, the learning rule used, agents behaviours, results and discussion.

2 Multi agent system description

The simulation model is composed by a 2 dimensional world bounded by walls. It contains two different objects: agents and food places. The agents, referenced by their position as $a_j(t)$, where a_j has 2 components (x,y coordinates indexed by $a_{j,x}$ and $a_{j,y}$), with $j = 1, .., N$. Agents move with a differential driving system (two wheels) [1]. Food places are disks located at fixed position $f_j(t)$ with $j = 1, ..., M$, and produce a variable quantity of food.

Agents have different sensors which enable them to sense obstacles, other agents presence and others internal state of satedness, at different ranges (proximal and distal see Fig. 1). Every object labeled with a certain index j produce a signal carried by a uniform potential field $G_{j,type}$ with a limited range, which is sensed by the corresponding sensor type (type can be avoid,food or sated). The signals from the proximal sensors (x_0) are originally used to drive the agents reflexes which can either be avoidance or attraction. The signals from the distal sensors are used for learning so that the agent is able to generate anticipatory reactions instead of the reflexes.

In the next section we are going to describe the learning algorithm enabling the agent to replace the reflexes with the predictive actions. Once learning algorithm is described, we will describe the different reflexes and possible anticipatory reactions.

2.1 ICO learning

The input correlation learning rule (ICO [11]) is a Heterosynaptic learning rule, it is unsupervised and performs a confounded correlation between a predefined reflex signal (x_0) and a reflex predicting signal (x_1). Hence, this learning

978-0-7695-3272-1/08 $25.00 © 2008 IEEE
DOI 10.1109/LAB-RS.2008.19

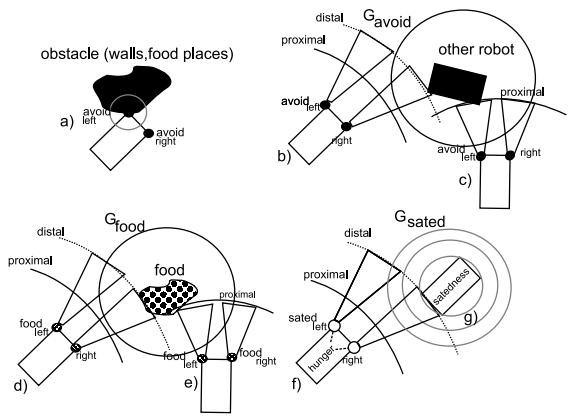


Figure 1. Overview of the different signals used in this simulation. Circles labeled with G (avoid,food,sated) represents uniform potential fields, circles on the robot's front are input sensor, cones irradiating from them represent the field of view of sensors, proximal and distal lines represent sensors range. Case a): an agent touches a food place or a wall with its proximal left avoidance sensor $avoid_{left,prox}$, a proximal signal is generated. Case b): an agent reads the potential field G_{avoid} produced by another, with its right distal sensor $avoid_{right,dist}$. Case c): an agent reads the potential field G_{avoid} produced by another agent, with both left and right proximal sensors $avoid_{left,prox}, avoid_{right,prox}$. Case d): an agent reads the potential field G_{food} produced by a food place, with its right and left distal sensors $food_{right,dist}, food_{left,dist}$. Case e): an agent reads the potential field G_{food} produced by a close food place, with both left and right proximal sensors $food_{left,prox}, food_{right,prox}$. Case f,g): an hungry agent f (with $Hunger = 1$) reads the satedness signal G_{sated} produced by the sated agent g.

algorithm identifies and exploits causalities between temporal sequential signals.

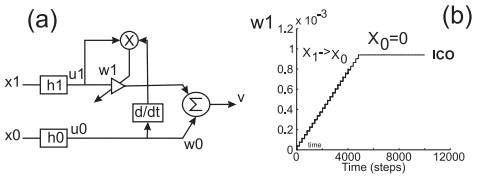


Figure 2. Figure (a) shows the ICO learning basic block composed by 2 inputs x_0, x_1 filtered by h_0, h_1 and the output v. Figure (b) shows the weight change of w_1 during time. At the beginning $w_1 = 0$, then for 5000 simulation steps x_1 anticipates x_0 and the w_1 grows until 1.0. After 5000 simulation steps reflex is suppressed $x_0 = 0$ and w_1 stabilize to $1 \cdot 10^{-3}$.

Fig. 2 shows the ico learning block which has two inputs x_0,x_1 from the agent's sensor that are filtered by bandpass filters h_0, h_1:

$$h(t) = \frac{1}{b} e^{at} sin(bt). \tag{1}$$

$$a = -\pi \frac{F}{Q}, b = Imp = \sqrt{(2\pi F)^2 - a^2}. \tag{2}$$

F is the oscillation frequency and Q the quality factor. The band passed signals $u_i(t)$ are transferred with weight w_i to the output neurons. In the output neuron the output $v(t)$ is calculated by summing up all incoming signals according to their weights:

$$v(t) = \sum_{k=0}^{1} w_k u_k. \tag{3}$$

which represents the input for the motor system. The unsupervised character of the ICO learning rule is reached by the synaptic weight w_1 to be adapted by the weight change rule:

$$\frac{d}{dt} w_1 = \mu u_1 \frac{du_0}{dt}. \tag{4}$$

The weight change is dependent on the derivative of the reflex input signal u_0, the input signal u_1 and a learning rate μ. The learning rule has been shown to be useful for avoidance and attraction mechanisms and has fast and stable convergence.

2.2 Neural controller and notation

For the sake of simplicity, the agent's neural controller is analyzed block by block according to the requested be-

haviours (avoidance and attraction) in Figs. 3,5,6. The core is composed by two ICO neurons, labeled with L-eft and R-ight, connected to the motor outputs left and right. Both ICO neurons have a constant bias input B with weight 4.0 that makes the robot move forward if inputs are absent. Rectangular block labeled with L,R are band pass filters, with parameters F, Q referred to equations 2,2. Synaptic weights of the ICO block in Fig. 2(a) are labeled with W capital letter and two pedex that indicates the weight type (predict is learned and reflex is fixed) and the synapse position (left,right). ICO neurons have recurrent synaptic connections, labeled as $W_{R2L}, W_{L2R}, W_{selfR}, W_{selfL}$ to implement a hysteresis effect, so that the controller doesn't instantly follow signals (see [3] and [2]). It means reactions on an incoming signal are time shifted. This is useful to enable agents to escape from acute angles: if an agent incurs in an concave acute angle and hasn't hysteresis, will get stuck inside, turning left and right alternatively.

3 Avoidance

Agents and walls are obstacles. Agents produce obstacle signals (see Fig.1 (a) for obstacles and Fig.1 (b),(c) for other agents). Every agent $a_j(t)$ has a potential field associated:

$$Gavoid_j(t) = G(x - a_{j,x}(t), y - a_{j,y}(t)). \quad (5)$$

that is sensed by the corresponding inputs of other agents $a_k(t)$ (with $k \neq j$) labeled as $avoid_{left,right}$. Walls and food places don't produce G_{avoid}, so that agents sense them using proximal signals that are generated by collisions: when 2 distinct agents j and l collide at time t_0, such that $||a_j(t) - a_l(t)||_2 < D$ (D is the radius of the agent) an impulse is produced $x_0(t_0) = \delta(t_0)$. The neural controller for the avoidance behaviour is described in Fig. 3 (see also [12]), every ICO neuron (left and right) has 2 corresponding reflexes (left, right short range sensors) connected:

$$x_{0,l}(t) = avoid_{prox,r}(t); x_{0,r}(t) = avoid_{prox,l}(t). \quad (6)$$

and 2 corresponding predictive signals (left, right long range sensors) connected:

$$x_{1,l}(t) = avoid_{dist,r,d_o}(t); x_{1,r}(t) = avoid_{dist,l,d_o}(t). \quad (7)$$

Connections between input synapses and ICO neurons (motor neurons) are negative to evoke a retraction. Predictive signals are used to learn to anticipate this behaviour from a longer distance. So $x_{0,l}(t), x_{0,r}(t)$ and $x_{1,l}(t), x_{1,r}(t)$ are regarded as motor error signals and $x_{0,l}(t), x_{0,r}(t)$ are the error signals for learning. When for example $x_{0,l}(t) > x_{0,r}(t)$ the neural controller produces a negative motor output for both $V_l < 0$ and $V_r < 0$ with $|V_l| < |V_r|$ until $x_{0,l}(t) = x_{0,r}(t) = 0$.

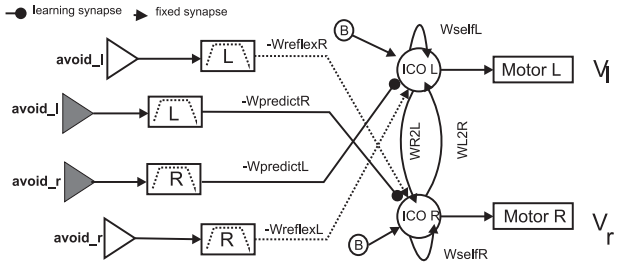


Figure 3. Avoidance network: ICO left and right are two neurons implementing the ICO learning rule, their output is a sigmoid and is connected (after normalization into $[v_{min}, v_{max}]$) to the motor speed commands. Grey triangles represents distal inputs, while white triangles represent proximal inputs. The learned synaptic weights are associated to the distal synapses (thick lines) while the fixed are associated to the proximal synapses (dotted lines). To produce a retraction behaviour left and right weights must be different such that $W_{predict,L} > W_{predict,R}$ and $W_{reflex,L} > W_{reflex,R}$, if $W_{predict,L} = W_{predict,R}$ and $W_{reflex,L} = W_{reflex,R}$ robot will just go back without turning.

4 Agents and satedness communication

Every agent has an internal state: hunger and its complementary satedness. Hunger is an exponential function of time:

$$H_{hunger}(t) = 1 - e^{(-(t/\tau_{starv}))}. \quad (8)$$

where τ_{starv} is the time to get hungry. Satedness signal is the inverted function of the hunger:

$$H_{sated}(t) = 1 - H_{hunger}(t). \quad (9)$$

Hunger and satedness functions are time shifted (internal state is resetted) at time t_b, $(H_{hunger}(t-t_b), H_{sated}(t-t_b))$, when an agent touches a food place:

$$t_b = t \Leftrightarrow |food_{l,prox,d_1}(t_b) + food_{r,prox,d_1}(t_b)| > \theta_F. \quad (10)$$

or touches a sated agent (see Fig. 4):

$$t_b = t \Leftrightarrow |sated_{l,prox,d_1}(t_b) + sated_{r,prox,d_1}(t_b)| > \theta_A. \quad (11)$$

and agent is touching an obstacle

$$t_b = t \Leftrightarrow |avoid_{prox,l}(t_b) + avoid_{prox,r}(t_b)| > \theta_O. \quad (12)$$

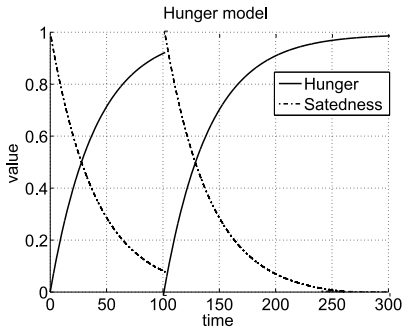


Figure 4. Hunger state in function of time. At time step 100 the agent touches a food place, thus its hunger state is resetted to 0 and so its complementary satedness to 1.

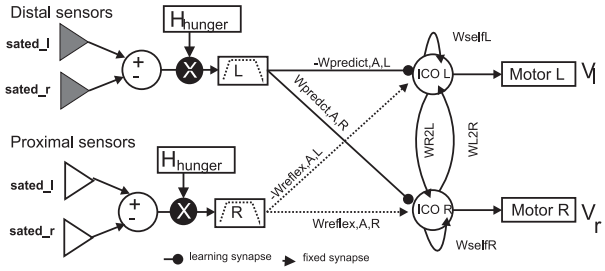


Figure 5. Attraction toward sated agents:two more inputs are added to the ICO neurons. Grey triangles represents distal inputs, while white triangles represent proximal inputs. The learned synaptic weights are associated to the distal synapses (thick lines) while the fixed are associated to the proximal synapses (dotted lines). Synaptic weights must be equal in module and opposite in sign $|W_{predict,A,L}| = |W_{predict,A,R}|$ and $|W_{reflex,A,L}| = |W_{reflex,A,R}|$ where $A = agent$. Hunger internal state is multiplied for proximal and distal input difference

where $\theta_F, \theta_A, \theta_O$ are thresholds for food, agents and obstacles respectively.

Satedness internal state of agent a_i, is transmitted to other agents $a_k(t)$ (with $k \neq j$) by means of a potential field (see Fig.1 (g)):

$$G_{i,sated}(t) = H_{sated}(t) \cdot G(x - a_{i,x}, y - a_{i,y}). \quad (13)$$

Agent $a_k(t)$ senses $G_{i,sated}$ (see Fig.1 (f)) with 2 reflexive inputs $sated_{l,prox,d_1}(t), sated_{r,prox,d_1}(t)$ whose difference feeds the reflexive input:

$$x_0(t) = sated_{l,prox,d_1}(t) - sated_{r,prox,d_1}(t). \quad (14)$$

and as predictives $sated_{l,prox,d_2}(t), sated_{r,prox,d_2}(t)$ whose difference feeds the predictive input:

$$x_1(t) = sated_{l,dist,d_2}(t) - sated_{r,dist,d_2}(t). \quad (15)$$

where $d_2 > d_1$. The internal state $H_{hunger}(t)$ is multiplied for $food(t)$ (left,right and proximal,distal) and $sated(t)$ (left, right and proximal, distal see Fig.5). Inputs for the attraction task are shown in Fig.5. So $x_0(t)$ and $x_1(t)$ are regarded as motor error signals and $x_0(t)$ is the error signal for learning. When for example: $x_0(t) > 0$ implies that a sated agent is on the left $sated_{l,prox,d_1}(t) > 0$, the neural controller produce $V_L < V_R$, agents turns left until $x_0(t) = 0$ that means either $x_0(t) = x_1(t)$ (a sated agent is in front) or $x_0(t) = x_1(t) = 0$ (no sated agent in front). An hungry agent is producing $G_{i,sated}(t) = 0$, therefore other agents will be repelled since it's emitting only the G_{avoid} signal. *Aggressive agents*: If we want make an agent more aggressive $H_{sated}(t)$ can be multiplied for the distal sensors $avoid(t)$ left, right (in Fig.3 the H_{hunger} block can be introduced after each of the grey triangles). It implies that when an agent is not sated, it will ignore the obstacle signal G_{avoid} produced by the other agent.

5 Food attraction

Every food place $f_j(t)$ with $j = 1, ..., M$ produces the signal (see Fig.1 (d),(e)):

$$G_{j,food} = G(x - f_{j,x}, y - f_{j,y}). \quad (16)$$

which is sensed by agents a_i by the inputs labeled as $food_{left,right}$ (see Fig. 1 (d),(e)). Every food place f_j contains a limited amount of food modeled by the variable q_j that is decremented every time an agent touches the food place at t_b: $q_j(t_b + 1) = q_j(t_b) - \theta_q$ ($\theta_q \leq 1$). When $q_j = 0$ the food place j is exhausted and food signal is suppressed $G_{j,food} = 0$. After a random period the food place is restored $q_j = 1$.

Required inputs for the attraction behaviour are introduced in Fig. 6 (see also [12]), for every ICO neuron an additional reflex is added:

$$x_0(t) = food_{l,prox,d_1}(t) - food_{r,prox,d_1}(t). \quad (17)$$

x_0 it's the difference between the left and right proximal food input sensors. A predictive input is added:

$$x_1(t) = food_{l,dist,d_2} - food_{r,dist,d_2}(t). \quad (18)$$

$x_1(t)$ it's the difference between the left and right distal food input sensors. Thus $d_2 > d_1$ such that the distal food

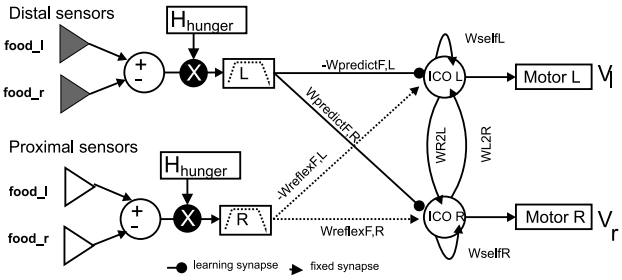


Figure 6. Attraction toward food:two more inputs are added to the previous network. Grey triangles represents distal inputs, while white triangles represent proximal inputs. The learned synaptic weights are associated to the distal synapses (thick lines) while the fixed are associated to the proximal synapses (dotted lines). Synaptic weights in the attraction task must be equals in module and opposite in sign: $W_{predict,F,L} = W_{predict,F,R}$ **and** $W_{reflex,F,L} = -W_{reflex,F,R}$ **where** $F = food$

sensor values are predictive on the proximal food sensors. So $x_0(t)$ and $x_1(t)$ are regarded as motor error signals and $x_0(t)$ is the error signal for learning. When for example: $x_0(t) > 0$ implies that food place is on the left, the neural controller produces $V_L < V_R$, agent turns left until $x_0(t)$ becomes 0.

6 Sub-system formation

To study the formation of subsystems we introduce the following measures [1]:

$$\delta_{w,agent} = \frac{|W_{predict,A}(0) - W_{predict,A}(t)|}{W_{predict,A}(0)}. \quad (19)$$

$$\delta_{w,food} = \frac{|W_{predict,F}(0) - W_{predict,F}(t)|}{W_{predict,F}(0)}. \quad (20)$$

Every sub-system has an integer counter $n_s(t), n_p(t)$ such that $n_s(t) + n_p(t) = N$. **If an agent is a seeker** which means $\delta_{w,agent} > \delta_{w,food}$ (it's more attracted by food places than other sated agents) the relative counter is updated $n_s(t+1) = n_s(t) + 1$. **If an agent is a parasite** which means $\delta_{w,agent} \leq \delta_{w,food}$ (it's more attracted by sated agents than food places) the relative counter is updated $n_p(t+1) = n_p(t) + 1$.

[1] $W_{predict,A}$ is the average of $|W_{predict,A,L}|, |W_{predict,A,R}|$: $W_{predict,F}$ is the average of $|W_{predict,F,L}|, |W_{predict,F,R}|$

7 Results

Sub-system formation is analyzed in function of a parameter γ pre-multiplied for the distal signals in eq. 15, so that the distinction between the adaptive and reflexive behaviour is shown:

$$x'_{1,l}(t) = \gamma \cdot x_{1,l}(t). \quad (21)$$

$$x'_{1,r}(t) = \gamma \cdot x_{1,r}(t). \quad (22)$$

$$0 \leq \gamma \leq 1. \quad (23)$$

The γ factor help us to understand how communication affects the sub-system formation:

- $\gamma = 1$: the agents learn using both proximal and distal signals
- $0 < \gamma < 1$: the agents learn using a reduced version of the distal signals
- $\gamma = 0$: the agents are reactive because only use reflex.

For our simulation, a population of $N = 20$ agents is provided with $M = 4, 10, 18, 20$ food places progressively. Population dynamic is observed for a duration $t > T_{sim} = 80000$ steps (about 800 seconds with a step $\triangle T = 0.01 seconds$). Figure. 7 reports the number of seekers $n_s(t)$ (thick line) and parasites $n_p(t)$(dotted line) in function of time. The population ratios of seekers to parasites ($r(T_{sim})$) at the end of the simulation for every case $M = 4, 10, 18, 20$ are respectively $r(T_{sim}) = 5/15, 8/12, 10/10, 12/8$. The case $M = 0$ is trivial: there are no food signals $G_{food} = 0$ in the world, therefore the agents only learn to avoid obstacles. For $M > 20$ seekers percentage increases, when $M > 28$ for $t > T_{sim}$ only seekers are present.

- with $\gamma < 0.1$: parasites are absent $n_p(t) = 0, \forall t > 0$ (also with aggressive configuration see 4). For $\gamma = 0$ is obvious because they are not learning to use the satedness signal.
- $0.1 \leq \gamma < 1.0$: parasitism is not a stable condition that means after a short period the initial condition is restored ($n_s = 0$) (data not shown)
- with $\gamma = 1$ parasitism is a quasi-stable condition. With scarce resources ($M = 4$) and $\gamma = 1$, after 600 seconds, the number of seekers (see Fig. 7 (a)) stabilize to 5. With abundant resources ($M = 18$), after 600 seconds, the number of seekers (see Fig. 7 (b)) stabilize to 10. After 800 seconds (data not shown) small oscillations around the stable point occur in both cases, suggesting that the system has reached an attractor.

- the population ratio between seekers and parasites $r(t \gg T_{sim})$ depends on the ratio between the number of robots and the number of food places N/M:
  - with scarce resources ($M = 4$) parasites are prevalent: $n_p(t \gg T_{sim}) = 15 \pm 1$, $n_s(t \gg T_{sim}) = 5 \pm 1$.
  - with abundant resources ($M = 18$) seekers and parasites are in dynamical equilibrium (oscillate around the stable point after T_{sim} steps): $n_p(t \gg T_{sim}) = 10 \pm 2$ and $n_s(t \gg T_{sim}) = 10 \pm 2$.

Moreover, if the number of agents are changed during simulation the system react promptly and stabilize to a new state. In Fig.8 seekers are stabilized in the range 4 ± 1, when 6 more agents are added at time 400, seekers re-stabilize in the range 7 ± 1. This property means that our system is robust to the perturbations of the environment.

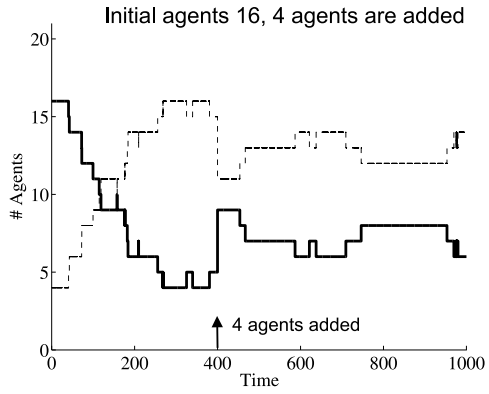


Figure 8. Simulation starts with 16 agents, then 4 agents are added after 400 seconds.

The main question is now: why one behaviour b_2 (agent sated attraction) prevails on the other b_1 (food attraction) when resources are scarce? In a previous work [10] it's introduced an information measure defined as PI which reflects the performance of the predictive learning. For every behaviour b_1, b_2, the agent is learning to predict the reflex: the larger the weights, the higher the value of the predictive information $PI(b_1), PI(b_2)$. Thus an agent using ICO tends to minimize the entropy measure at its input through the environment's loop. When the system is in the steady state condition, it can be regarded as a strategic game where every agent has 4 choices:

- no action: $b_1 = b_2 = 0$
- go for the food:$b_1 = 1, b_2 = 0$
- go for another sated agent:$b_1 = 0, b_2 = 1$

Table 1. Table resuming foraging performance over 100 simulations

Ratio N/M	20:4	20:18
$\gamma = 0$	$F_{tot} = 320 \pm 2$	$F_{tot} = 230 \pm 2$
$\gamma = 0.1$	$F_{tot} = 245 \pm 2$	$F_{tot} = 231 \pm 2$
$\gamma = 0.5$	$F_{tot} = 284 \pm 2$	$F_{tot} = 248 \pm 2$
$\gamma = 1$	$F_{tot} = 439 \pm 2$	$F_{tot} = 340 \pm 2$

- go for both: $b_1 = 1, b_2 = 1$

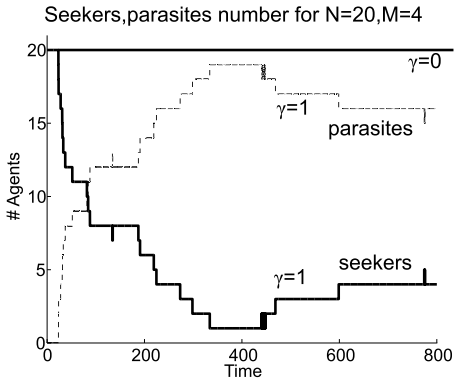
Agents must make their decision according to a common shared resource: food. Agents always make non neutral choices: the no action case is excluded because they need food. The strategic game can be considered as a summarization game [4]: each agent is influenced by the actions of all others, but only through a global summarization function. Every agent's payoff is an arbitrary function of their own action and the value of the summarization funcion, which is determined by the population play. It can be proven that for this game a Nash equilibrium exists in the space of joint mixed strategies. What is relevant for our model is that if every agent chooses $b_1 = 1, b_2 = 1$ it will have the maximum entropy (and not the minimum as wanted). The best strategy which minimize the entropy in average for every agent is to have separate choices: $N/2$ agents choose $b_1 = 1, b_2 = 0$ and the other $N/2$ choose $b_1 = 0, b_2 = 1$. Indeed when the number of food places is equal to the number of agents, agents self distribute between the 2 choices or sub-systems. When food resources become few, the majority of agents prefer $b_1 = 0, b_2 = 1$ (becomes parasites) because the entropy around the food zones is increased due to the high competition.

7.1 Food performance

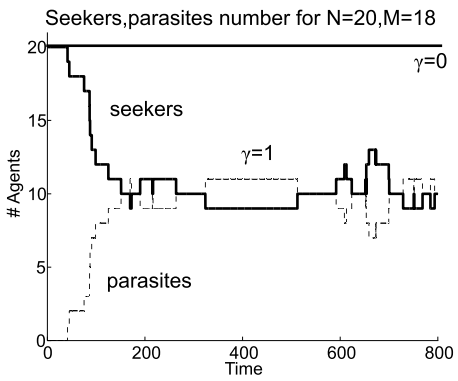
To analyze the performance of the system in consuming food: the number of the total bites $F_{tot} = F_{seek} + F_{parasite}$ is considered. F_{seek} is the number of total times that agents touched food places (eq. 10) and $F_{parasite}$ is the number of total times that agents touched other sated agents (eq. 11). Table 1 shows the foraging performance (average $\pm$ range) over 100 simulations and it can be noticed that with $\gamma = 1$ the best performance is achieved in terms of food foraging.

8 Discussion

There are three basic learning techniques applied to multi agent systems (MAS): reactive, logic-based and so-



(a) Case considering scarce resources: with $\gamma = 0$ only seekers are present (top line is constant to 20 agents), with $\gamma = 1$ sub-system formation is achieved and parasites become more than seekers after 200 seconds



(b) Case considering abundant resources: with $\gamma = 0$ only seekers are present (top line is constant to 20 agents), with $\gamma = 1$ communication sub-system formation is achieved and after 200 seconds they are in dynamical equilibrium

Figure 7.

cial. Our agents are reactive. In reactive systems the overall behaviour emerges from the interaction of the component behaviours. Since internal processing is avoided: agents respond to the changes of their environment in a timely fashion. In Q-learning [15], reactive agents are given a description of the current state and have to choose the next action so as to maximize a scalar reinforcement received after each action. The task of the agent is to learn from indirect, delayed reward, to choose sequences of actions that produce the greatest cumulative rewards. Reinforcement learning algorithms attempt to find a policy that maps states of the world to the actions the agent ought to take in those states. In economics and game theory, reinforcement learning is considered as a boundedly rational interpretation of how equilibrium may arise. Reinforcement models for MAS suffers of 2 disadvantages:

1. complexity may be exponential in the number of environmental states.

2. discrete models: agents choose from a set of actions in a discretized world

This problem is crucial in MAS: when an agent is learning the value of its actions in the presence of other agents, it's learning in a nonstationary environment. In [16] it's introduced a novel exploration strategy for the Q-algorithm applied in a predator-prey game to obtain convergence. Unfortunately the cost of communication in their model is very expensive, so they discuss only the case without communication. Our unsupervised rule is computationally efficient (since it doesn't rely on states) and inspired on the evidence that organism tends to mantain a weak homeostasis with the environment (see [7]).

In [13] Q-learning is applied in a predator-prey game, where agents cooperate in different ways. It's interesting to analyze the communication method called sharing sensation. The model is composed by 1 hunter, 1 prey and a scouting agent. At each step, the scout sends its action and sensation back to the hunter: the hunter relies initially on his sensation and then on the scout's sensation. Therefore the scout can be compared in our model as the distal signal: hunter can see the prey at longer distances. Performance (number of steps to capture a prey) is then compared in the 2 cases: scouting vs no scouting. Performance is superior in the scouting case as well as in our model performance it's superior when agents use both proximal and distal signal (see table 1).

Another approach used to develop communcation in MAS is in [6] where a genetic-algorithm is used to evolve 1000 robots (divided in 100 colonies). The control system used is a feed-forward neural network with 10 inputs and 3 output neurons. The network was encoded using a genetic string of 240 bits. Synaptic weights are only evolved and not learned and robots has a sensory-motor cycle of 50 ms. Our controller makes use of 12 inputs (2 more) and 2 motor neurons + 1 the $G_{satedness}$ signal, but doesn't use a sensory-motor cycle allowing fast responses. Thus [6] makes use of genetic selection and recombination to produce robots behaviours (communication strategies). Hence the agent system don't self organize in classes. In our model the system subdivides in 2 groups of agents specialized in different tasks: one group gets the food and distributes it (seekers) and the other one (parasites) collects it. Other similar works, making use of evolved recurrent neural networks (RNNs) are reported in [9] and [8]. In [8] a population of agents evolved for the ability to solve a collective naviga-

tion problem develop individual and social/communication skills. A particular evolved behaviour resembles our system differentiation: "a differentation of the modalities with which communication is regulated (... e.g. specialized asymmetrical interaction forms in which one robot acts as a speaker and one robot acts as an hearer)". [9] studies the evolutionary adaptivity of RNNs to varying environmental conditions, such as the number of interacting robots. However, the communication strategy was robust only for small changes. In our model, agents adapt continuously to the environment, they self-organize efficiently with varying robot number N and varying M food places (Fig.8). Moreover our system converges to a quasi-stable state thanks to the stability provided by the learning rule used. Further work will focus on different food signalling strategies: agents can signal the food presence honestly or dishonestly and how those strategies affect the performance and the sub-system formation. The theoretical information framework based on the input entropy minimization and decision game theory is being validated against different test cases.

References

[1] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. Bradford, Colorado, 1984.

[2] P. Huelse. Dynamical neural schmitt trigger for robot control. *In Dorronsoro,J.R.*, ICAN 2002(2):83–90, 2002.

[3] P. Hülse, Wischman. Structure and function of evolved neuro-controllers for autonomous robots. *Connections Science*, 16(4):249–266, March 2004.

[4] M. J. Kearns and Y. Mansour. Efficient nash computation in large population games with bounded influence. In *UAI*, pages 259–266, 2002.

[5] N. Luhmann. *Social Systems*. Stanford University Press, Stanford, California, 1995.

[6] D. F. S. M. S. Magnenat. Evolutionary conditions for the emergence of communication in robots. *Current biology*, 17(17):514–519, March 2007.

[7] D. J. McFarland. *Intelligent behavior in animals and robots*. MIT Press, Cambridge, MA, 1993.

[8] D. M. S. Nolfi. Origins of communication in evolving robots. *From Animals to Animats*, (9):789–803, September 2006.

[9] S. W. F. Pasemann. The emergence of communication by evolving dynamical systems. *From Animals to Animats*, (9):777–788, September 2006.

[10] B. Porr, A. Egerton, and F. Wörgötter. Towards closed loop information: Predictive information. *Constructivist Foundations*, 1(2):83–90, 2006.

[11] B. Porr and F. Wörgötter. Strongly improved stability and faster convergence of temporal sequence learning by utilising input correlations only. *Neural Computation*, 18(6):1380–1412, 2006.

[12] K. Stamm. Individual learning and the dynamics in predator-prey populations. *Göttingen informatic journal*, (ZFI-NM-2007-08):243–259, April 2006.

[13] M. Tan. *Multi-Agent Reinforcement Learning: Independent vs. Cooperative Learning*. Morgan Kaufmann, San Francisco, CA, USA, 1997.

[14] P. Verschure and T. Voegtlin. A bottom-up approach towards the acquisition, retention, and expression of sequential representations: Distributed adaptive control III. *Neural Networks*, 11:1531–1549, 1998.

[15] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[16] O. H. Z. W. Z. W. X. Xiaoming. A novel multi-agent q-learning algorithm in cooperative multi-agentsystem. *Intelligent Control and Automation. Proceedings of the 3rd World Congress*, 1:272 – 276, 2000.