An immune network approach for self-adaptive ensembles of autonomic components: a case study in swarm robotics

Nicola Capodieci[1], Emma Hart[2] and Giacomo Cabri[1]

[1]University of Modena and Reggio Emilia
[2]Institute for Informatics and Digital Innovation, Edinburgh Napier University
[1]{nicola.capodieci, giacomo.cabri}@unimore.it
[2] e.hart@napier.ac.uk

Abstract

We describe an immune inspired approach to achieve self-expression within an ensemble, i.e. enabling an ensemble of autonomic components to dynamically change their coordination pattern during the runtime execution of a given task. Building on previous work using idiotypic networks, we consider robotic swarms in which each robot has a lymph node containing a set of antibodies describing conditions under which different coordination patterns can be applied. Antibodies are shared between robots that come into communication range facilitating collaboration. Tests in simulation in robotic arenas of varying complexity show that the swarm is able to learn suitable patterns and effectively achieve a foraging task, particularly in arenas of high complexity.

Introduction

Current and emerging ICT scenarios increasingly rely on complex distributed software systems in order to function properly in dynamic and unpredictable environments Zambonelli et al. (2011). This results in a need for the software controlling such systems to become *autonomic* in adapting behaviours such that quality of service of the system is maintained.

In Zambonelli et al. (2011), the authors describe two important dimensions of adaptation that can occur within autonomic systems, which they refer to as *where* and *what*. *Where* relates to where adaptation takes place, i.e. at the individual or ensemble level. *What* on the other hand refers to the set of mechanisms the system can utilise to adapt. They distinguish between *self-adaptation* and *self-expression*: the former refers to components or ensembles modifying their parameters so as to exploit their current abilities, whereas the latter describes the ability of radically modifying at runtime the structure of components and ensembles. In terms of ensembles, self-expression could result in re-structuring in terms of topology (e.g., switching from a hierarchy to a collective of peers) or of control regime for interactions (e.g., switching from being a collective decision-making ensemble to a competitive market-based one) Zambonelli et al. (2011).

While self-adaptation mechanisms at both individual and ensemble levels has been the focus of much research (e.g. see Salehie and Tahvildari (2009) for overviews), mechanisms for achieving self-expression in distributed systems are less well-understood, particularly with respect to systems that enable software within an ensemble to express at run-time the most useful interaction topology or control regime. Inspired by the fact the features and properties apparent in the natural immune system such as scalability, adaptivity through learning and decentralization map naturally to those desired in autonomic systems Cabri and Capodieci (2013), we describe an idiotypic-network approach to self-expression. We consider a swarm-robotic scenario in which multiple robots have to fulfil a simple foraging task. Multiple *coordination patterns*, i.e. collaborative strategies, are available to the robots to face the problem; robots contains a lymph node describing a set of antibodies that indicate a suitable strategy — by sharing of antibodies across the swarm when robots come into contact with each other, the entire swarm is able to learn to solve the problem over time, even when placed in environments of varying complexity.

Previous Work

A fruitful line of work within robotics that started with Ishiguro et al. (1995) has applied inspiration from Jerne's idiotypic network theory to develop behaviour arbitration mechanisms in individual robots. Antibodies consist of a matching condition to match environmental conditions, an action, and of receptors that enable interactions with other antibodies. The resulting network of stimulatory and suppressive connections alters concentrations of antibodies; the one with the highest concentration applies its action. Various weaknesses in this work that required hand-coding of antibodies for instance have recently been addressed in Whitbrook et al. (2010b), who consider evolutionary methods for generating antibodies and reinforcement learning to connecting them in a network, resulting in a system that has been ported successfully to real-robots Whitbrook et al. (2010a). We extend this work in that we deal with *swarms* of robots rather than individuals, and that rather than considering individual actions, the robots must select a cooperative strategy to take
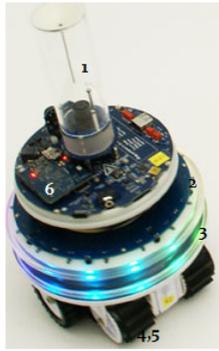
Figure 1: A footbot. Sensors and actuators are placed as follows: (1) omni-directional sensing camera, (2) proxity sensors, (3) LEDs, (4,5) wheels and ground sensors, (6) RAB sensor and actuator

Sensor	Purpose
LED actuator	12 light emitting diodes surround the robot. LEDs can be different colors
Omni-directional light sensing camera	senses colored lights: returns their distance and angle of perception with respect to the sensing robot
Wheels actuator	enables movements
24 proximity sensors	for detecting collisions
Range and Bearing sensor and actuator (RAB)	sends/receives infra-red packets of a fixed 10 bytes size
Ground sensor	detect strong variations in the floor color

Table 1: Sensors and actuators modelled in robot simulation

Figure 2: Simple example arena. Footbots in action are shown, together with the blue light symbolizing the nest and, in the opposite end, a red light for indicating the food area.

part in.

There have also been previous attempts to apply idiotypic network ideas to swarms. In Jun et al. (1999), a distributed version of Ishiguro et al. (1995) is proposed, in which individual robots choose an appropriate action according to a changing environment and the experiences of nearby robots. In Luh et al. (2006), instead of a single action, more complex behaviours are selected as a results of modelling a two layered immune network, merged as interaction among antibodies in the single robot and then distributed throughout all the swarm. Our works differs from both these publication, firstly in the robotic scenarios used and for switching *coordination patterns* as a response of the designed artificial immune network; this implies not only selecting behaviours, but also roles, statuses and interactions within the coordination pattern, and can result in dynamically changing interacting topologies, e.g. switching from a peer-to-peer coordination pattern to a purely stigmeric communicative collaborative effort and vice-versa.

Task and robot description

We model a simple task in which a swarm of robots, initially randomly distributed in a confined arena, are required to collect food from a source and return that food to a nest, in an iterative process. The goal of the task is to maximise the total amount of food returned to the nest in a fixed time period, although performance can also be evaluated in terms of the rate of food collection over shorter time periods. The robot used in the simulation is called a *footbot*, and is one of the types of robot used in the swarmanoid project Dorigo et al. (2012) and in other previous research regarding swarm robotics coordination, e.g. Capodieci and Cabri (2013) — see Figure 1. The sensors and actuators used on the robot are relatively simple. For instance, each robot is not able to calculate its position in space, nor does it have any concept of orientation. The modelled sensors and actuators are shown in table 1.

Coordination Patterns

Three coordination patterns are considered: a completely swarm approach, a peer-to-peer (p2p) directly communicative approach and a baseline coordination strategy operated with the limited amount of sensors and actuators. The artificial immune system (AIS) enables the robots to select between these patterns based on their assessment of the complexity of their current environment and expected utility.

RACO: Robotic Ant Colony Optimization

RACO is a robotic application of the well-known ant colony optimization family of algorithms Dorigo et al. (1996). In a typical ACO implementation a virtual pheromone trail is used to guide agents towards an object of interest; in a robotic scenario, as it is difficult for a robot to actively modify the environment by laying pheromones, some of the robots in the swarm adopt the role of pheromone: each robot
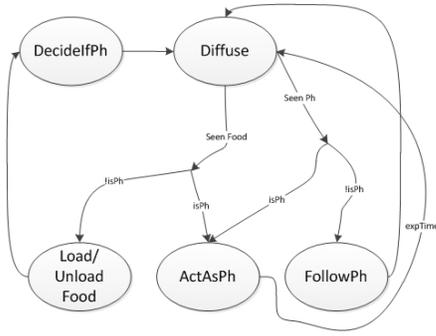
Figure 3: States and conditions for the RACO algorithm

Figure 4: Exponential evaporation time for the pheromone robot: emitted light intensity A over time

is assigned a (small) probability p of acting a pheromone, with the remaining $(1 - p)$ act as ants. All the robots start by uniformly diffusing into the area following a very simple diffusive algorithm: each robot starts with a random orientation and starts moving in a straight line; if a collision is sensed in one or more of the 24 proximity sensors the robot reacts by moving in the opposite direction from the sensed collision. If a robot that has decided to act as a *pheromone* reaches the food area (represented by a red emitting light) or the nest area (blue light), the robot stops moving and lights all 12 of its LEDs in yellow. Robots from the *ants* group continue to diffuse uniformly until they sense a pheromone; at this point they follow the pheromone trail rather than diffuse uniformly. The *intensity* of the pheromone is represented by the brightness of the emitted light. The more intense the light, the greater the distance from which it can be perceived by other robots. Intensity decays according to equation 1 which models evaporation over time (also shown in Figure 4).

$$Intensity(t) = MaxIntensity$$
$$- \left(\frac{\beta e^{\frac{6t}{expTime}}}{1 - (\alpha - \beta)} + e^{(\alpha - \beta)\left(\frac{6t}{expTime}\right)} \right) \quad (1)$$

In eq. 1, *MaxIntensity* represents the maximum value for the light intensity emitted by the robot's LEDs. *expTime* is the maximum expiration time (when the intensity reaches zero), *t* is the time variable and α and β are constants. When the expiration time of the pheromone reaches zero, the *pheromone* robot switches its role to become an *ant* robot once more. However, if an *ant* robot comes within a distance d of the pheromone, the pheromone regenerates and its value is reset to *MaxIntensity* . When the expiration time of the pheromone reaches zero, the pheromone robot switches its role to become an ant robot (see Figure 3). Every robot makes its decision to act as a pheromone every time the food or nest area is reached, and that's how new pheromones are created.
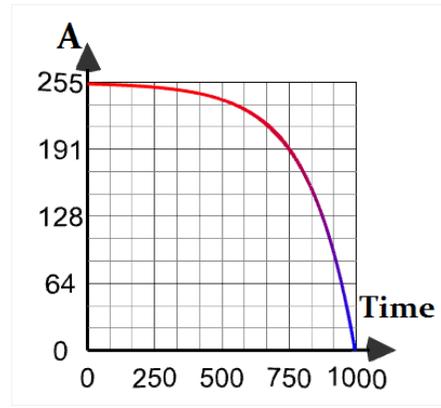
AMORPH: Amorphous computing inspired path formation

The second coordination pattern strongly relies on communications in the form of packets sent and received with the RAB sensor and actuator built in each robot. This is purely peer-to-peer (p2p) approach in which roles are assigned dynamically according to the current situation. This strategy is inspired by the work of *Abelson et al.* in Abelson et al. (2000) which studies the use of bio-inspired algorithms for achieving collaboration among a potentially large number of devices connected in unknown way. The algorithm is described in detail in Abelson et al. (2000); the modifications required to adapt this to a robotic swarm are discussed below. Its underlying concept is to diffuse a gradient across the amorphous net of communicating robots indicating the shortest path from the nest to the food source.

The swarm starts by contracting who will be the path opener: any robot sensing the nest area (blue light) sends a message to nearby robots indicating its distance from the light. Any robot receiving a distance greater than the one it currently holds begins to diffuse; the process results in the single robot that is furthest from the light becoming the path-opener and remains stationary (robot A in figure 5). Robots diffuse uniformly until one reaches the nest area (red light) at which point it becomes the *path-closer* (robot-F in figure 5) and halts. This robot sends out a packet to nearby robots via infra-red signalling through the RAB actuator) consisting of the tuple *(SenderID, gradient, state, successorID)*. *Gradient* is an indicator of the distance to the start of the path, state is a boolean value indicating whether a robot is currently part of the path, *successorID* is the ID of the robot that previously sent the highest gradient value to the receiving robot. The path-closer sends the maximum gradient value (255). Receiving robots store the highest value received g. They then transmit a new gradient $g - 1$. The chain ends when the *path-opener* receives a gradient and transmits
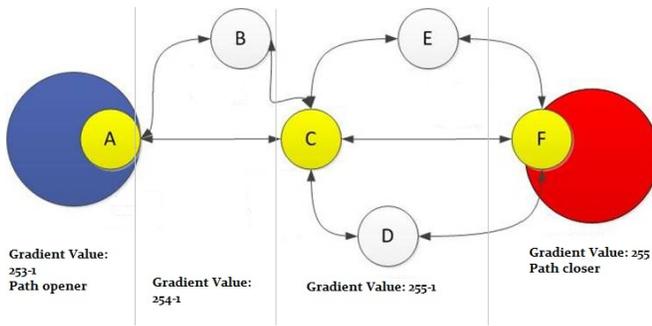
Figure 5: Schematic of an example situation in the amorphous path formation algorithm: communication is variable due to random placement, obstacles and limited RAB range for signalling. A possible communication topology is shown by the lines and arrows. At the end of the algorithm, robots A,C,F provide the path from food to nest area.

a message that a path is ready to be formed (setting the *state* packet to 1). All robots (including the *path-opener*) receiving state 1 set their LEDS to a specific colour thus indicating the path. Robots not on the path now follow the lit path to the food.

An emergent property of this algorithm is that the path of lit robots formed is exactly one-robot thick. The robots that are inside this path will be referred as *nodes*, while the other robots can still be called ant robots since their behaviour now is very similar to the one described in the previous coordination pattern.

Baseline strategy: blind diffusion

An additional cooperative strategy is represented by a coordination pattern that uses the minimum amount of sensors and actuators. It is called blind diffusion, since all the robots are *blind*, in the sense that they disable the omni-directional camera, thus are unable to sense lights and colour for being attracted towards the point of interests in the arena.

All robots apply a simple diffusion algorithm: when a collision is sensed through the proximity sensor, they steer in opposite direction by rotating in a direction opposite to the angle to incidence w.r.t. their body centre of the proximity sensor detecting the collision. An emergent property of this algorithm is that the swarm becomes uniformly distributed throughout the whole arena. In this algorithm, food is sensed by ground sensors built in the wheels of each robot that detect floor colours based on a grey-scale. However, it should be clear that this algorithm will be much less effective that the previous approaches in terms of the amount of food collected.

Remarks on coordination pattern performance

Previous experimentations with each of three coordination patterns in a small area led to a number of observations
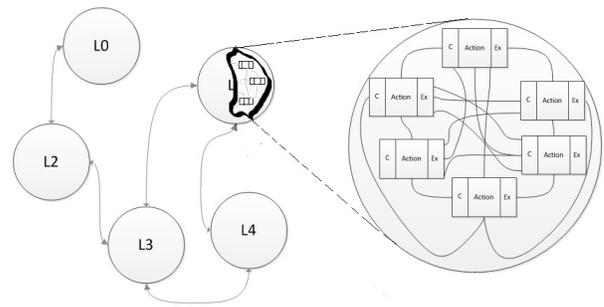
Figure 6: lymph-nodes are robots with variable connectivity and they can host a multitude of interconnected antibodies.

that motivates the selection strategy introduced in this paper. Briefly, it was observed that the AMORPH algorithm performs best (in terms of total food collected) in complex arenas, where complexity relates to both the size of the arena and the number of obstacles present. In simple (obstacle free) arenas, RACO performs best — probably due to the fact the the RACO algorithm enables more robots to be devoted to foraging, rather than being tied up in path construction. Both algorithms struggle in some situations as both lack a proper obstacle avoidance behaviour, therefore certain configurations of either node or pheromone robots can impede performance. This motivates the algorithm presented, which is not only able to dynamically alter the current coordination pattern due to online feedback regarding current performance, but also results in subsets of the swarm in fact following different coordination patterns at any given moment. This is described in the next section.

SelfEx - Model

The model introduced extends work originally discussed in Ishiguro et al. (1995). Significantly, we extend the concept of controlling an individual robot through a network of antibodies that match conditions to actions to a distributed swarm in which the robots only have limited communication range. Each robot is modelled as a lymph node that contains a set of antibodies, connected in an idiotypic network, see Figure 6. Robots can diffuse antibodies from one lymph node to another, thus the contents of lymph nodes are continually adapted. In contrast to previous work, antibodies determine the *coordination pattern* that should be executed by the robot at a given moment rather than a specific action. The algorithm (**SelfEx**) is given in listing 1 and is described in detail below.

As we can see in fig. 6 each robot is modelled as a lymph-node in a net of lymph-nodes whose connectivity varies according to the ever changing position of the robots. By taking a look inside each robot, we can see how each lymph-node can host a multitude of interconnected antibodies and each antibody is characterized by a variable concentration value and it is divided into three main parts that will be ex-

plained in the following section.

Algorithm 1 SelfEx: overview

1: **Initialisation**: each robot initialised with a set of antibodies
2: **Maturation**: each robot estimates the complexity of the environment
3: **Selection**: each robot selects a start coordination pattern based on its antibody that highest affinity with the environment
4: **repeat**
5:    **Evaluation & Affinity Update**: every $eval$ timesteps, each robot evaluates its performance and updates the affinities between antibodies in its own idiotypic network
6:    **Concentration Update**: each robot updates the concentration of each antibody within its lymph node
7:    **Diffusion**: the best antibodies are diffused to robots within communication range
8: **until** stopping criteria

Antibodies and Affinity An antibody is a tuple of values $\langle$ Conditions, Action, expected utilities $\rangle$ as follows:

- **Condition** has two variables, $complexity$ and $status$. $Complexity$ is a real-value representing the complexity of the arena. $Status \in \langle 0, 1, t \rangle$ where 0 indicates an ant robot in the RACO model, 1 a node robot in the AMORPH model and t indicates the time a robot has spent in the pheromone state in the case of pheromone robots.

- **Action** $\in \langle BlindDiffusion, RACO, AMORPH \rangle$.

- **Utility** is the value of the food collected in a time period, $eval$ by each robot

The *affinity* between an antibody and the current environment is calculated as the Manhattan distance between the calculated *complexity* and the *complexity* value stored in the antibody.

Initialisation A set of antibodies to seed networks were derived from an initialisation phase; during this phase, a swarm utilised each individual coordination pattern in each tested arena over a 50000 time-step period. During this period, each robot logged its status, perceived complexity index and the amount of food it managed to collect during the evaluation time. These results were then averaged to build an initial set of 10 seed antibodies. Each robot's lymph node is loaded with the same set of 10 antibodies found. The initial concentration of each antibody in each lymph node is set to zero.

Maturation The purpose of the maturation phase is for each robot to estimate the complexity of the environment. This is calculated according to algorithm 2. Note that the lower the value of $complexity$ calculated, the higher the complexity of the arena.

Algorithm 2 SelfEx: Maturation Phase

1: initialise lymph node of each robot with a set of antibodies
2: Goal $\in \{Food, Nest\}$
3: **repeat**
4:    **for all** robot i in $\mathcal{R}$ **do**
5:      Move according to *blind diffusion pattern*
6:      **if** Goal encountered **then**
7:        $maturationCounter_i \leftarrow maturationCounter + 1$
8:      **end if**
9:    **end for**
10: **until** end of maturation phase
11: $complexity$ = averaged value of each robots $maturationCounter$

Evaluation and Affinity Update After $eval$ timsteps, each robot evaluates its own performance u and compares this to the expected utility u_E indicated in the currently active antibody. If $u > u_E$ then positive feedback is given *to* the selected antibody by increasing the affinities *from* all other antibodies towards the selected one; in addition to that, the expected utility field of this antibody is updated with the newly obtained value. If $u < u_E$ then negative feedback results in the the selected antibody increasing its affinity to all other antibodies. Affinity values r (between antibody i and j) are calculated as follows:

$$r_{i,j} = \omega |obtUtilities - expUtilities| + \frac{K_0}{|dAc - abAc|}$$

$$\omega = \begin{cases} status & \text{if } status \text{ is } \leq 1 \\ 1 - \frac{status}{evalTime} & \text{if } status > 1 \end{cases}$$

(2)

The difference between obtained utilities (*obtUtilities*) and expected utilities (*expUtilities*) is weighted according to the $status$ variables (see section). In addition, the affinity is adjusted according to the difference in the detected area complexity and the complexity value stored in the antibody, regulated by the constant K_0 (< 1).

Concentration Update Antibody concentration updates are performed in a similar manner to that of Ishiguro et al. (1995) as shown in eq. 3. The main differences are that we assume that every antibody i is connected to all the other antibodies, antibodies with low concentration are not removed, and we do not add new antibodies.

$$\frac{\mathrm{d}c_i}{\mathrm{d}t} = K_1 \sum_{j=0}^{N} r_{j,i} c_i c_j - K_2 \sum_{k=0}^{N} r_{i,k} c_i c_k + K_3 D c_i \quad (3)$$

The first term of eq. 3 refers to the stimulation part of the immune net; the second term (with a negative sign) refers to the suppression operated by the other antibodies and the latter term takes into account the euclidean distance D among all the conditions of the antibodies to the obtained/detected values of status, utilities and area complexity. Three constants ($K_{1,2,3}$) regulates the contribution of each of these effects. The resulting concentration is than squashed to fit the $[1...255]$ range.

Diffusion The calculated concentrations of the four antibodies with highest concentration in each lymph node are now shared amongst other robots in range, in order to distribute knowledge throughout the ensemble regarding current performance in a single packet[1].

Ant robots *broadcast* such packets but do not *receive* packets as they are able to form their own estimate of how much food has been collected. *Pheromone* and *Node* robots average its concentration with the external concentration sent by other robots for each antibody. Following the updating step, the robot selects the antibody with the highest concentration and executes the coordination pattern indicated.

Experiments and simulations

The pre-experimentation phase to establish the initial set of antibodies was undertaken in two areas; the first was characterized by the highest complexity index (visible in fig. 2, the simplest area), while the second arena with a high-index used an extensive sized hexagonal arena with an obstacle between the straight path from nest to food area. Tests with the main algorithm were performed in three different arenas: the two described above and one of intermediate complexity. Each experiment was repeated 10 times, with a different random initial distribution of robots each time. In each arena, experiments were performed with the single RACO and AMORPH algorithms as well as with SelfEx. Parameters were tuned through an empirical process — values used in the results reported are shown in table 2. The experiments were performed using *ARGoS* swarm robotics simulator (Pinciroli et al. (2012)).

Results and remarks

As stated previously, the experiments were evaluated on the basis of total performances over the total length of the experiment and the variation of food collected each evaluation period (by the entire swarm). This latter metric is useful to note the gradual improvements of each collaborative effort:

---
[1] Only four antibodies are distributed due to the limitations of the RAB sensor

Parameter	Description	Value
eval	evaluation period for affinity step	10,000
l	total timesteps experiment run	200,000
m	maturation phase length	5500
K_0	from equation 2	0.05
K_1	from equation 3	0.25
K_2	from equation 3	1
K_3	from equation 3	0.05
p	probability to become a pheromone	0.35
$expTime$	from equation 1 (RACO)	1500
α	from equation 1 (RACO)	0.3
β	from equation 1 (RACO)	0.8

Table 2: Values of parameters used for the simulations

the AIS approach should show an average increasing trend in the amount of food collected in each time interval, in order to indicate the network is *learning*. Figure 7 shows the results of the simplest arena. SelfEx and RACO have similar performance, although SelfEx is less variable than RACO. AMORPH performs best in terms of the median value of food collected and variability. However it is clear from the right hand figure that the performance of AMORPH deteriorates over time, whereas SeflEX and RACO show steady improvement, suggesting that given further time they may at least equal the AMORPH performance. In the second arena (Figure 8) RACO performs best. SelfEx performs better than AMORPH, particularly in terms of variability. As in the first arena, the performance of SelfEx increases over time, a pattern not clear in RACO and AMORPH. Finally, in the third and most difficult arena, the best performance is achieved by SelfEx (Figure 9). In this arena, performance does not increase over time, suggesting the ensemble reached consensus on the coordination pattern to adopt very early in the experiment.

Figure 10 shows the percentage of robots in each evaluation period that follow the different coordination patterns in Arenas 1 and 3 (figures from a single run). In Arena 1, the swarm is divided in its choice of pattern — interestingly, some robots still select the blind diffusion strategy. In contrast, in Arena 3, robots eventually converge towards a single coordination pattern. The ability of SelfEx to enable heterogeneous behaviours might be particularly useful in more dynamic environments.

Conclusions and Future Work

We have introduced a novel approach to managing a collaborative swarm of robots by taking inspiration both from autonomic computing and bio-inspired algorithms, and extending previous work that used idiotypic network models to control individual robots. The model was tested on a simple foraging task in which robots had to choose a coordination pattern from a set of three different possibilities. Each robot
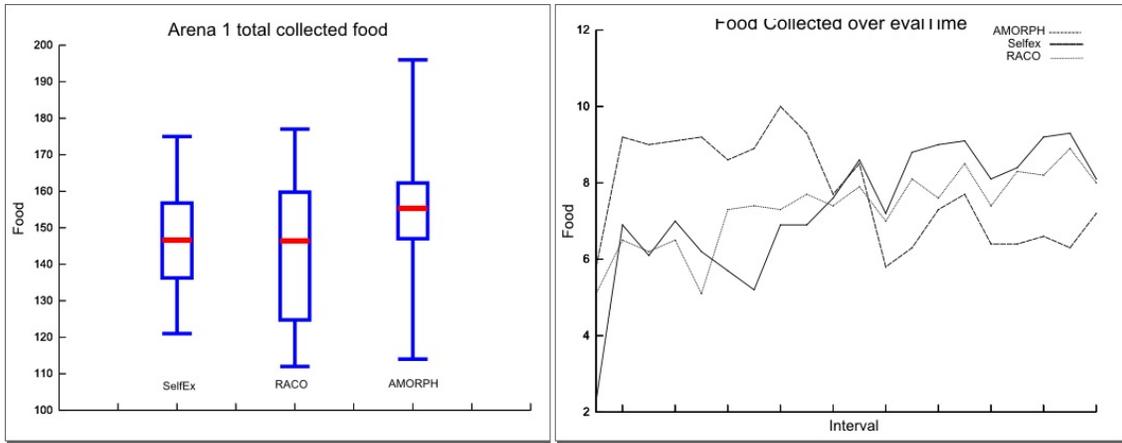
Figure 7: First test arena results
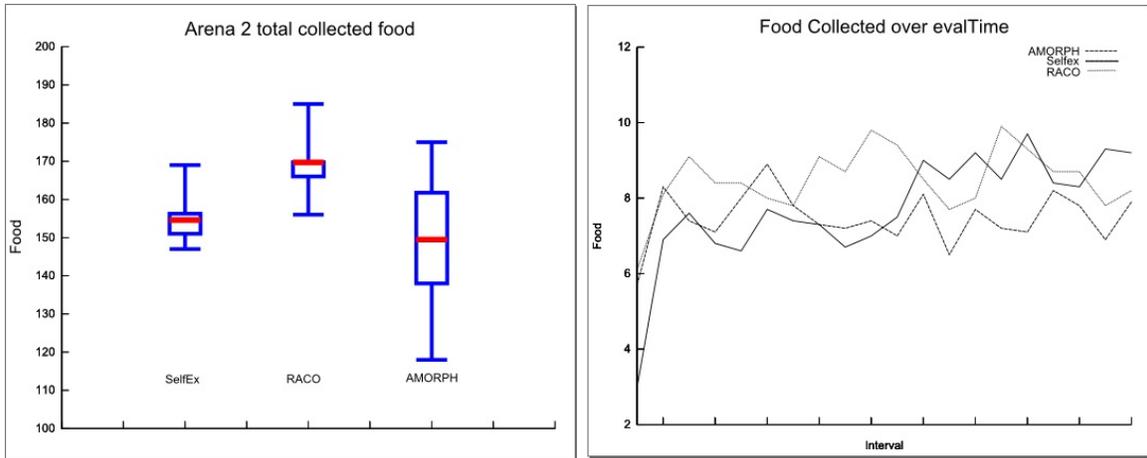
Figure 8: Second test arena results
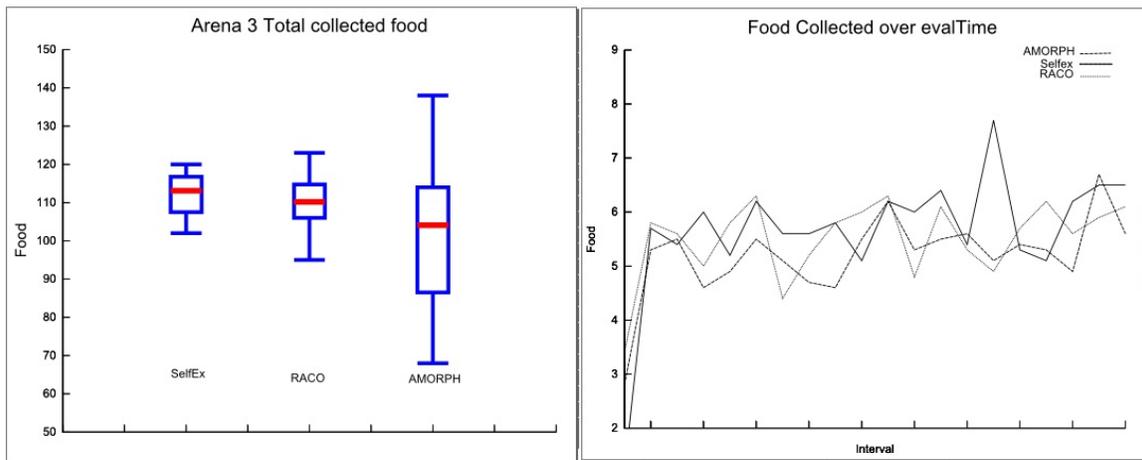
Figure 9: Third test arena results

Figure 10: Distrubution of coordination patterns amongst robots in each time interval. The top figure shows Arena 1, the lower Arena 3. The figures are 'stacked' in the order blind diffusion,RACO, AMORPH from the bottom

is a modelled as lymph node hosting a connected network of antibodies. Sharing of antibodies across robots enables common decisions to be made as to the best manner to fulfill the task. The results shown that the ensemble was able to deploy Self-Expression, i.e. fragmenting into sub-sets and initially choosing different coordination pattern. The results show that the SelfEx approach was able to learn and improve its performance over time, thus demonstrating the cognitive ability as emerging property of an immune net. However, the overall performances also show that how our approach is slow to converge and the maturation phase implies smaller quantities of collected food during the initial time intervals: there is a large room for improvements by appropriately tuning the parameters involved. Additional improvement could be gained by using a hyper-mutation process in order to optimise individual antibodies within a network via a local search process.

References

Abelson, H. et al. (2000). Amorphous computing. *Communications of the ACM*, 43(5):74–82.

Cabri, G. and Capodieci, N. (2013). Runtime change of collaboration patterns in autonomic systems. In *In Proc. of IEEE international Conference on Advanced Information Networking and Applications (AINA–2013) FINA 2013 workshop*. IEEE.

Capodieci, N. and Cabri, G. (2013). Collaboration in swarm robotics: a visual communication approach. In *in proc.of The 2013 Conf. on Collaboration Technologies and Systems (CTS 2013), The 4th International Collaborative Robots and Human Robot Interaction workshop*, page to appear. IEEE.

Dorigo, M. et al. (2012). Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics and Automation Mag.*

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29–41.

Ishiguro, A., Watanabe, Y., and Uchikawa, Y. (1995). An immunological approach to dynamic behavior control for autonomous mobile robots. In *IEEE–RSJ International Conference on Intelligent Robots and Systems 95. Human Robot Interaction and Cooperative Robots*, volume 1, pages 495–500. IEEE.

Jun, J., Lee, D., and Sim, D. (1999). Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system. In *in Proc. of the IEEE International Conference on Systems, Man, and Cybernetics. IEEE SMC '99*, volume 6, pages 614–619. IEEE.

Luh, G.-C., Wu, C.-Y., and Liu, W.-W. (2006). Artificial immune system based cooperative strategies for robot soccer competition. In *Strategic Technology, The 1st International Forum on*, pages 76–79. IEEE.

Pinciroli, C. et al. (2012). Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence, Springer Berlin*, 6(4):271–295.

Salehie, M. and Tahvildari, L. (2009). Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 4(2):14.

Whitbrook, A., Aickelin, U., and Garibaldi, J. (2010a). Real-world transfer of evolved artificial immune system behaviours between small and large scale robotic platform. *Evolutionary Intelligence*, 3(3):123–136.

Whitbrook, A., Aickelin, U., and Garibaldi, J. (2010b). Two-timescale learning using idiotypic behaviour mediation for a navigating mobile robot. *Appl. SoftComput*, 10(3):876–887.

Zambonelli, F., Bicocchi, N., Cabri, G., Leonardi, L., and Puviani, M. (2011). On self-adaptation, self-expression, and self-awareness, in autonomic service component ensembles. *in Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on.*, Ann Arbor, Michigan, USA: IEEE Computer Society:108–113.