Comparison of genetic algorithms used to evolve specialisation in groups of robots

Tomassino Ferrauto[12], Gianluca Baldassarre[2], Gabriele Di Stefano[1], Domenico Parisi[2]

[1] Dipartimento di Ingegneria Elettrica e dell'Informazione, Università dell'Aquila
tomassino.ferrauto@istc.cnr.it, gabriele@ing.univaq.it

[2] Laboratory of Autonomous Robotics and Artificial Life, Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerche (LARAL-ISTC-CNR)
gianluca.baldassarre@istc.cnr.it, domenico.parisi@istc.cnr.it

Abstract

This paper investigates the role of genetic algorithms in determining which kind of specialisation emerges in decentralised simulated teams of robots controlled by evolved neural networks. As shown in previous works, different tasks may be better solved by robots specialized in a particular manner. However it was not clarified how much the genetic algorithm used might drive the evolution of one kind of specialisation or another: this is the goal of this paper. The study is conducted by evolving teams of robots that have to solve two different tasks that are better accomplished by using different types of specialisation (innate versus situated). Results suggest that the type of genetic algorithm employed plays a major role in determining how robots specialize and in most of the cases the algorithms used tend to always yield the same specialization. Only one of the algorithms tested led to the emergence of the most suitable kind of specialisation for each one of the two tasks.

1 Introduction

The field of collective robotics, or multi-robot systems, is receiving an increasing attention within autonomous robotics (for extensive reviews and taxonomies of multi-robot systems and of the tasks that can be tackled through them, see [1], [2], [13]. The goal of this paper is to start to systematically study the different types of specialisations and the role of genetic algorithms in determining which kind of specialisation emerges in different environmental conditions.

The robots of the multi-robot systems studied here have the following properties: (a) they have to collaborate to accomplish a common task ("cooperative tasks" are the most studied in the field, see [13]); (b) have a distributed control system (there are no "leader robots" or centralised controllers within the system, cf. [6]); (c) are guided by feed-forward memory-less neural-network controllers evolved with genetic algorithms (no learning process during the tests); (d) have no explicit communication (coordination has to rely upon perception, physical interactions, implicit communication, cf. [12]).

Multi-robot systems are important for engineering purposes since they have a number of strengths if compared to single robots: (a) some tasks can be carried out only, or with more advantages, through multi-robot systems [2]; (b) when they can be used, distributed controllers are simpler and more robust than centralised controllers in the case of failure of one element of the system; (c) systems that do not need to communicate are usually simpler, cheaper, and robust.

An important issue studied within the field of cooperative collective robotics is "specialisation". "To specialise" means "to concentrate on and became expert in a particular subject or field" (Oxford Dictionary). In adaptive multi-robot systems, specialisation can take place at least under two different drives: 1) the whole task of the team requires that the robots engage in different behaviours to be accomplished: this case is widely studied in cooperative collective robotics [14]; 2) the whole task of the team would not strictly require specialisation, but the robots still tend to assume and maintain specific roles (e.g., as it will be shown below, the robots that have to approach a light while staying in group tend to assume and maintain a specific spatial position with respect to each others even if this is not required by the task). This tendency is quite common in adaptive systems as it decreases the computation burden for each robot. Indeed, playing specific roles greatly diminishes the complexity of the set of input patterns to which the robots have to associate suitable actions (cf. [3], [6]).

Referring to bio-inspired adaptive controllers, such as neural networks, and embodied multi-robot systems, it is possible to define the following types of specialisation (notice that they might co-exist both at the level of the single robot or at the level of the team):

Body specialisation: different robots exhibit different behaviours due to their different sensors, actuators and bodies.

Innate specialisation: robots behave differently on the basis of controllers that differ from the beginning of the accomplishment of the task. This type of specialisation can emerge when genetic techniques are used to evolve the controllers, and different genomes, or different parts of a genome, are used to encode the architecture and/or weights of the robots' controllers [4].

Learned specialisation: the robots develop a different controller (usually the weights of it), and hence different behaviours, on the basis learning algorithms [15].

Memory specialisation: the robots exhibit different behaviours on the basis of different value of internal memory units ([5], learned and memory specialisation might be considered as one type of "ontogenetic specialisation" based on changes of "internal states" of the controller during the task.

Situated specialisation ([6]): robots with identical controllers play different roles, and maintain them in time, on the basis of the different input patterns experienced. In this case, roles are usually allocated on the basis of initial random differences.

An important topic studied in the literature [14], and closely related to specialisation, is the "task allocation problem". The problem, that refers to the coordination mechanisms that the team uses to allocate different roles to its members, is particularly important for this study since, as we shall see, the emergence of the different types of specialisations in the multi-robot systems studied here was driven by the need to suitably manage the initial allocation of roles among the robots and the switching of roles during the execution of the task when this was advantageous.

The controllers used in this research were evolved with genetic algorithms. The evolution of the controllers of multi-robots systems presents several advantages in comparison to hand-designing them: 1) from an engineering perspective, evolution is a powerful method to design controllers when indirect and highly dynamical causal chains separate the controller's properties from the desired behaviours or behavioural outcomes. In fact, in these cases it would be difficult to envisage such causal chains in order to hand-design the controllers, while evolution first generates/modifies the controllers randomly and then selects them *a-posteriori* on the basis of their overall performance (this difficulty, quite common in autonomous robots, cf. [3], is even more impairing in multi-robot systems where the behaviour of the whole group strongly depends on the dynamical interactions between the robots, cf. [6]); 2) from a scientific perspective, the evolution of robot's controllers more likely generates insights on animals' behaviour, both in general it might suggest the possible paths followed by evolution in nature, and in particular because it might enlighten the functioning of specific mechanisms behind natural evolution.

In order to study the emergence of specialisations and how much the particular genetic algorithm drives evolution toward different kinds of specialisation two different tasks were designed: a collective light approaching task and a coordinated motion task. As shown below these two tasks are quite different as each one is better accomplished by teams exhibiting a particular kind of specialisation. For each task we evolved teams using different genetic algorithms: a) a genetic algorithm that uses different parts of a genome to encode the controllers of the robots of a team, and considered the team as the unit of selection (group selection); b) a genetic algorithm that creates teams by randomly drawing robots from a single population and that then selects for reproduction single robots instead of a whole team; c) a genetic algorithms that creates teams by randomly drawing robots from two separated population and that then operates selection at robot level within each population (cf. [7]). For both one-population and two-populations algorithms (i.e., the algorithms described in b) and c)), two versions were used: in one teams are randomly created at the beginning of the generation and then fixed for the entire generation, in another teams are shuffled several times in a single generation.

In the following sections the experimental setup and the results obtained will be described. Section 2 will describe in detail the tasks and the experimental environment as well as the first two genetic algorithms. After that in Section 3 the relationship between tasks and specialisation will be analysed and differences between the two different setups will be highlighted. Finally, in Section 4 the results of evolution with different genetic algorithms will be presented.

2 The experimental setup

The research presented here has been done in a simulated environment into which highly realistic simulated teams of robots have been evolved. The simulations presented here differ both in environmental conditions the teams of robots live in and in the task they are requested to perform.

2.1 The light-approaching and coordinated-motion tasks

In the light approaching task, the robots had to approach a light target while staying close to each other. The environment used for this task was composed of a rectangular arena of

1 by 2 m surrounded by walls (**Figure 1** shows the simulated version of the arena). Two halogen light bulbs of 230 W each were present in the arena, and were located in the middle of the west and east shorter walls at a height of 1.5 cm from the ground.

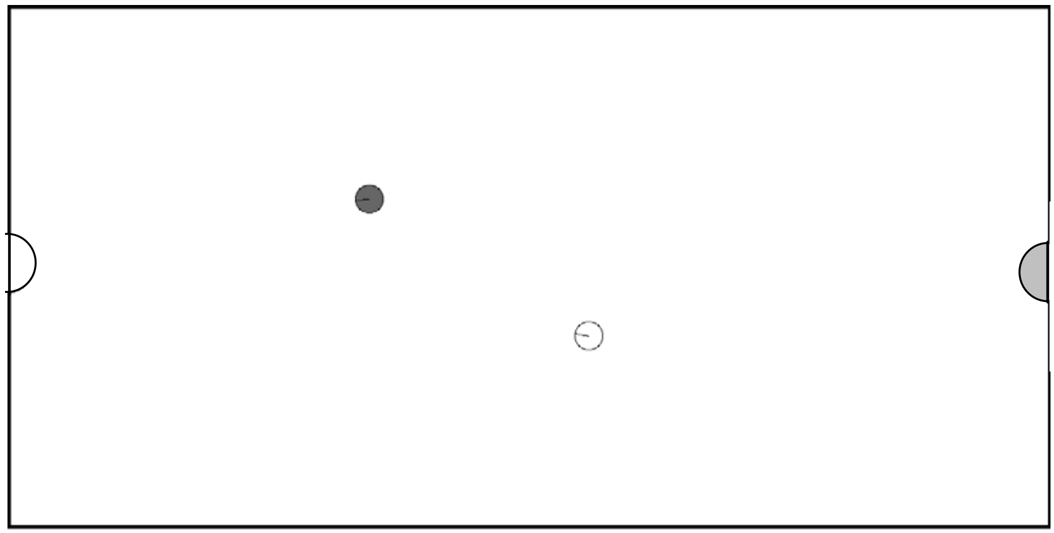

Figure 1: The arena with two robots and two light bulbs (light and grey semicircles: only one light was on at each time). The segment on each robot indicates the side with six sensors (see description of the robots reported below). The segment also indicates the direction of motion of the wheels.

The coordinated motion task was carried out with an open arena without any light source. In this task, the robots had to move as far as possible from the initial position while staying close to each other.

2.3 The robots and the neural controller

Two simulated Khepera^{TM} robots were used in all experiments (Figure 2, cf. [8]). The robots were provided with 8 infrared sensors, used to detect the presence of walls and other robots up to a distance of 45 mm, four light sensors, used to detect the light target up to 4 m (there were obtained using infrared sensors in passive mode), and four directional microphones, used to detect the position of other robots (these sensors, not present on the real robots, were simulated as described below). Each robot had two motors, each controlling the speed of one of the two wheels for motion, and a loudspeaker that continuously emitted a sound with fixed amplitude and a frequency that randomly varied within a given range, used to signal the own position to other robots (this actuator, not present on the real robots, was simulated as described below). Notice that microphones and loudspeakers allowed robots to detect each other at greater distance than infrared sensors. The groups of robots used in the experiments were composed of two robots with identical physical structure.

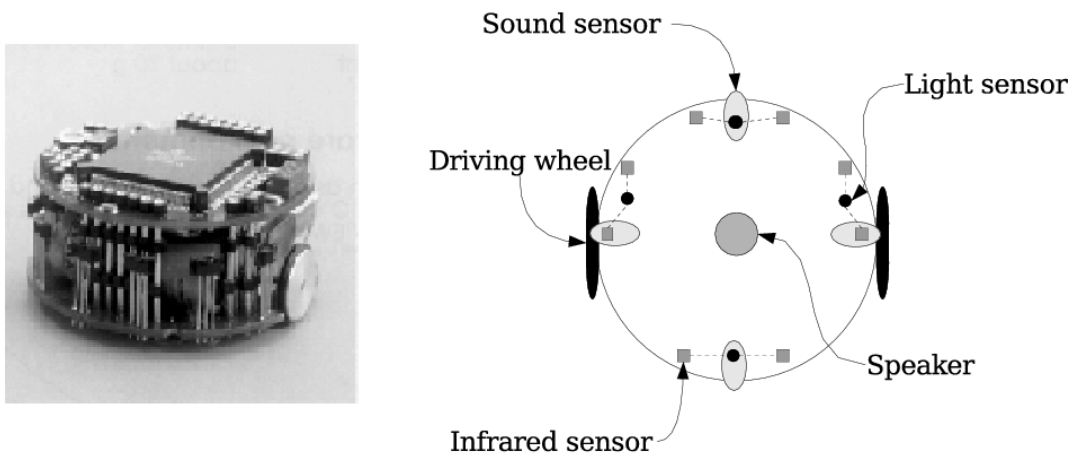

Figure 2: Left: The Khepera robot, whose radius measures 27.5 mm. Right: scheme of the robot. Dashed lines connecting a light sensor with two infrared sensors indicates that the value of that light sensor is obtained as the average of the activation of the two corresponding infrared sensors used in passive mode

To achieve a greater level of realism of the simulations, the data obtained from a sampling procedure carried out with the physical robot were used to compute the activation state of the infrared and light sensors (cf. [9]). This sampling procedure set a physical robot in front of walls, another robot, or halogen light, and measured the activation of the robot's infrared sensors (used in a passive way in the case of light perception) at different angles and distances. A geometrical simulation of shadows was also implemented to have more realistic activation of light sensors. In order to simulate the effects of different activations of the motors, the change of orientation and displacement in space of a physical robot was sampled in correspondence to different commands issued to the motors themselves (the sampled data relative to the infrared sensors activated by the halogen light were those used in a previous research, see [6]; the sampled data relative to the infrared sensors activated by walls and other robots, and the sampled data relative to motors, were those used in [10]).

As the physical robots were not endowed with direction microphones and loudspeakers, these were simulated as follows (cf. [6]). The sound amplitude A of sound in space was computed as:

$$A = \frac{1}{1 + \left(\frac{D_t}{1000} \right)^2} AF$$

where D_t is the distance of the microphone from the sound source, in millimetres, and AF is a scaling factor that simulates the effects of the microphone's orientation with respect to the sound source. AF was computed as:

$$AF = 1 - 0.9\frac{\alpha}{180}$$

where α is the convex angle, in degrees, between the direction pointed by the microphone and the direction of the sound source. The amplitude actually perceived by the microphone, PA, was computed as:

$$PA = \frac{2}{1 + e^{-A}} - 1$$

where A is the amplitude of the sound emitted by the other robot of the team, calculated using the previous formula (the simulator assumed that it was possible to filter out the activation of the microphone due to own sound).

To further increase the realism of the simulation, a random value with uniform distribution over ±0.05% was added to all sensors at each step.

In the light approaching task, each robot was controlled by a neural network (**Figure 3**) having 16 input neurons, each corresponding to a particular sensor of the robot, and a bias neuron. These neurons were directly connected with the 2 output neurons, whose value controlled the speed of the two wheels.

The activation of the output neurons was mapped onto the wheels' speed. If the activation value of an output neuron was between 0 and 0.5 the corresponding wheel rotated backward at a speed proportional to the activation value of that neuron, if it was 0.5 between 0.45 and 0.55 the wheel was still, if it was between 0.5 and 1 the wheel rotated forward.

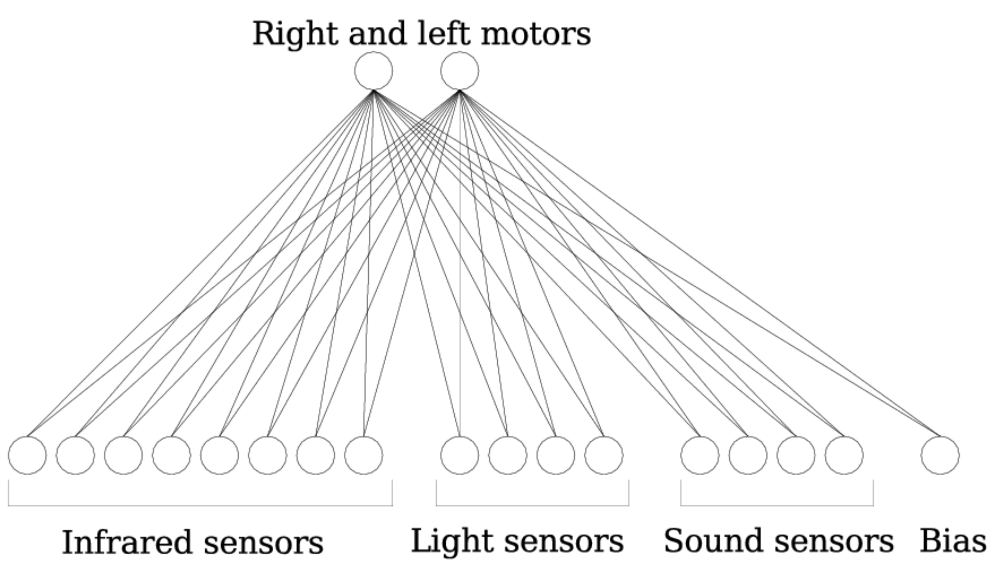

Figure 3: The neural network controlling robots in the first simulation

In the coordinated motion task, the robots used only the infrared sensors, and so the controllers had only 18 weights.

2.2 The fitness functions

In the light approaching task, each team was tested for 4 epochs which lasted 1500 steps each. During each step (which lasted 100 ms) the activation of robots' sensors was computed and sent to the controller, the controller calculated the activation of the output units, and finally this activation was used to move the robots. At the beginning of each epoch the robots of a team were randomly placed in the arena with random orientation and only the light on the left part of the arena was turned on. Then robots were left free to move and when the barycentre of the team reached a distance lower than 300 mm from the light currently on, this light was turned off and the other was turned on. Given this setup, the team had to change direction and move toward the turned on light to achieve a good performance.

To reward teams whose robots were able to move toward light while staying close to each other, a fitness function made up of two different components was designed: a group compactness component (GCC) measuring the ability of robots to stay close together, and a group speed component (GSC) measuring the ability of robots to move fast toward the light. The group compactness component was computed at each step t as follows:

$$GCC_t = 1 - \frac{D}{600}$$

where D was the distance between the barycentre of the two robots. If D was greater than 600 mm, the GCC for that step was set equal to 0. The group speed component was computed at each step t as follows:

$$GSC_t = \frac{1}{2}\left(1 - \frac{\Delta GD_t}{7}\right)$$

where ΔGD_t was the variation of the distance between the team's barycentre and the light target at step t and 7 was a constant equal to the maximum possible displacement of a robot in one step. Given this formula, GSC was less than 0.5 if a team moved away from light, 0.5 if it was still, and greater of 0.5 if it moved toward the light target.

The total fitness of a team was computed as the average of the two components over its whole life:

$$F = \frac{1}{M}\sum_{t=1}^{M}\left(\frac{1}{2}GCC_t + \frac{1}{2}GSC_t\right)$$

where M was the total number of steps of each team's life (M=6000).

In the coordinated motion task, each team was tested for 40 epochs which lasted 150 steps each. At the beginning of each epoch, the robots of the team were placed in the arena with random orientations and a distance of 15 mm between them. At this distance the robot could detect each other through the infrared sensors (when the two robots were more than about 35 mm apart, equal to the range of the infrared sensors, they could not perceive one each other).

The fitness function rewarded teams that were able to leave as fast as possible from the starting position while staying close to each other. Similarly to the light approaching task, the fitness was composed of two components: a group compactness component (GCC), measuring the ability of robots to stay close together, and a group speed component (GSC), measuring the ability of robots to move far away from the starting point. The group compactness component was computed at each step t as follows:

$$GCC_t = 1 - \frac{D}{300}$$

where D was the distance between the barycentre of the two robots. If D was greater than 300 mm, both GCC and GSC for that step are equal to 0 (notice that this formula is more demanding, in terms of group compactness, with respect to the formula used for the light approaching task: this was needed to evolve teams of robots capable of not loosing the perceptual contact between them). The group speed component was computed at each step t as follows:

$$GSC_t = \frac{\Delta GD_t}{7}$$

where ΔGD_t is the variation of the distance between the team's barycentre and the starting point at step t and 7 is a constant equal to the maximum possible displacement of a robot in one step.

As in the light approaching task, the total fitness of a team was computed as the average of the two components over its whole life:

$$F = \frac{1}{M} \sum_{t=1}^{M} \left(\frac{1}{2} GCC_t + \frac{1}{2} GSC_t \right)$$

where M was the total number of steps of each team's life (M=6000).

3 The tasks and specialisation

As stated above the two tasks used here differ in what kind of specialization is more suitable to accomplish them. As shown in [11] the light approaching task was more suitable for teams that exhibit situated specialisation. In fact, to achieve high fitness teams should be able to change direction as quick as possible; teams that use situated specialization can do that changing their orientations of about 180° by rotating on the spot (Figure 4). It is important to note that by doing this the two robots change their role in the team: for example the robot at the left side with regard to the light is now at the right or the robot in the front part of the formation is now in the rear part. On the contrary, when a team that uses innate specialisation is required to change direction because the light currently on is switched off and the other is switched on, its robots need to rearrange their position in the formation so that each one always plays the same role (Figure 4). To do this some time is required during which robots don't move towards the light target.

In [11] was also shown that the coordinated motion task is better accomplished by teams exhibiting structural specialisation. The main characteristic of this task was that roles need to be allocated at the beginning and are then fixed for the entire duration of the test. Because of the absence of an external marker (like the light in the previous task), teams that use situated specialisation find a lot of difficulties in allocating roles when both robots perceive the same input pattern. As shown in (Figure 5) they exhibit a good behaviour only when the starting orientations allows robots to assume different roles, e.g. when they have the opposite orientation with respect to the centre of the group. On the contrary, when the robots have symmetrical orientations with respect to the centre of the group, they have difficulties in allocating different roles. Instead, teams that use innate specialisation can easily solve the problem of the initial allocation of roles. Figure 5 shows that, at the beginning of the task, each of the two robots quickly manages to assume a specific position and orientation within the group, independently of the initial conditions.

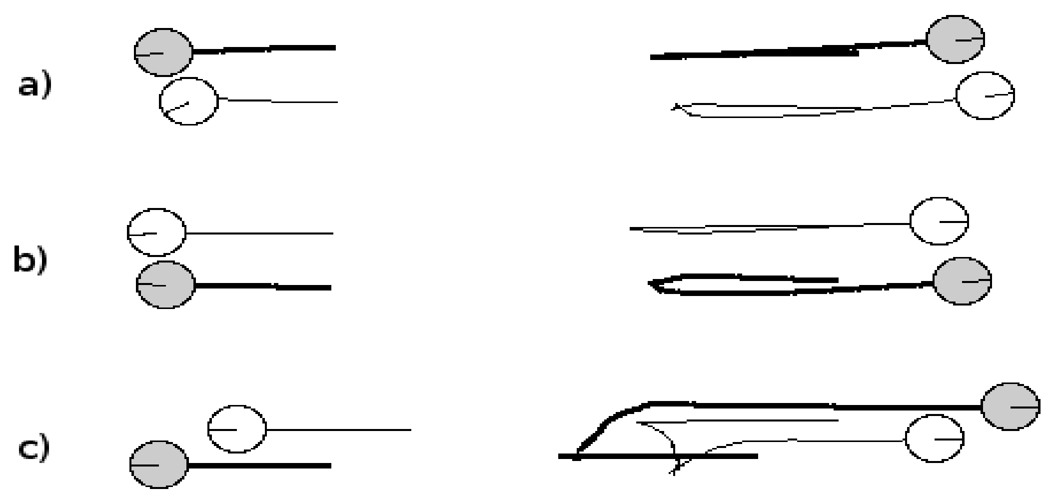

Figure 4: Qualitative behaviour of robots of teams that use situate and innate specialisation in the light approaching task, when the light target changes position. a) and b) show the behaviour of a team that use situated specialisation, c) shows the behaviour of a team that uses innate specialisation. The left part of the figure shows positions before the change of light position (light at left), while the right part shows the positions after the change of direction (light at right)

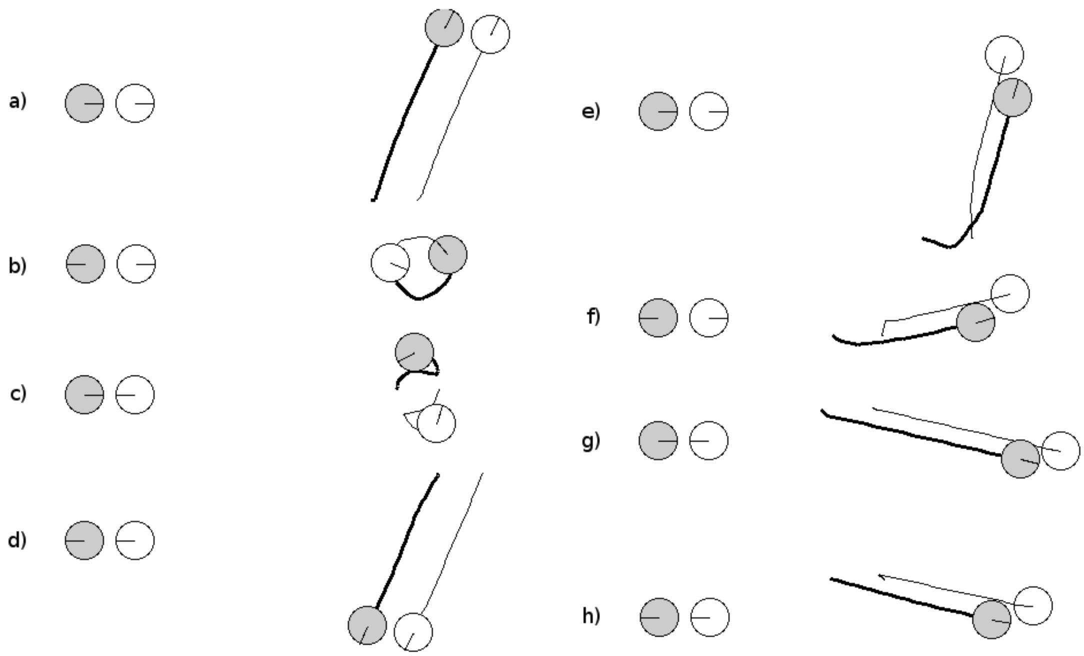

Figure 5: Qualitative behaviour of robots of teams that use situate and innate specialisation at the beginning of the coordinated motion tests. a), b), c) and d) show the behaviour of an homogenous teams, while e), f), g) and h) show the behaviour of an heterogeneous teams.

4 Influence of Genetic Algorithms on specialisation

In all algorithms evolution of controllers was replicated ten times starting from different initial genotype populations. In each replication, evolution lasted 600 generations. In both tasks the best 20 genotypes (in the algorithm that operates selection at team level) or 40 genotypes (in the algorithms that operate selection at robot level) of each generation were reproduced by generating five copies of each. During reproduction, each weight encoded in the genotype was mutated, with a probability of 5%, by adding it a random value in the range [-10.0, 10.0].

4.1 Selection at team level

In this algorithm the selection unit has been the whole team. The initial population consisted of 100 randomly generated genotypes and each genotype encoded the connection weights of the neural networks controlling the robots of a team. To allow, at least theoretically, both innate and situated specialisation to emerge, the genotype was made up of two parts, each corresponding to the weights of the neural network controlling a single robot of the team, so it encoded 68 different weights. The substantial difference with the clone approach, in which a single genotype is evolved and than the same controller is given to all robots of the team, is that now it is possible to have different controllers and so innate specialisation.

The analysis of evolved teams in both tasks show that innate specialisation emerges. However it is present not only in the coordinated-motion task, which “require” it, but also in the light approaching task, leading to sub-optimal performances (citare paper workshop precedente).

	Light-approaching task	***Coordinated-motion task***
Team selection (genotype made up of two parts)	Innate Specialisation: YES	Innate Specialisation: YES
Population: single Team creation: at beginning of each epoch	Innate Specialisation: NO	Innate Specialisation: NO
Population: single Team creation: only at beginning of each generation	Innate Specialisation: NO	Innate Specialisation: YES
Population: double Team creation: at beginning of each epoch	Innate Specialisation: YES	Innate Specialisation: YES
Population: double Team creation: only at beginning of each generation	Innate Specialisation: YES	Innate Specialisation: YES

Table 1: *Algorithms influence on the emergence of innate specialisation.*

4.2 Selection at robot level – one population

In this two algorithms the unit of selection has been the single robot and the fitness assigned to each of them depended on the fitness of the teams it has belonged to during its life. The population was made up of 200 randomly generated genotypes and each genotype encoded the connection weights of the neural network controlling a single robot. Depending on the task, the genotype encoded 34 or 18 weights as real numbers (see above). In the first algorithm 100 teams were formed at the beginning of each epoch by randomly coupling the robots of the population and, at the end of that epoch, the fitness of each robot was incremented by half the fitness of the team it belonged to (at the beginning of the generation all robots had fitness equal to 0). This way the same robot took part in several teams during its life. The second algorithm was similar to the first one, but new teams have been created at the beginning of each generation and then fixed for the entire generation. This implies that even if the selection unit is the robot, the elements of a team are both allowed to reproduce or both discarded.

The innate specialisation had difficulties to emerge especially when the teams were rearranged at the beginning of each epoch (second row of Table 1). In fact if, say, two different roles tended to emerge, individuals had 50% of chances to join a companion with the same specialisation, and, as consequence, the fitness function penalized them.

In order to measure the lack of innate specialisation in a population, we performed correlation measures among the genotypes of the population in such a way that the genotype of a single individual has been correlated with the genotypes of all other individuals. A strong correlation among all the genotypes of a population points out the absence of specialised individuals. In Figure 6 we show a typical result of a correlation measure after a seed of the first algorithm which arose in both the performed tasks. All individuals are represented in the X axis, ordered by their correlation value with a randomly chosen individual, in the Y axis the same individuals are present in the same order, finally, in the Z axis the correlation value between the genotypes of two individuals is plotted. The generalized high correlation value, very close to 1, is the main indicator of a population without specialized robots. A careful analysis of correlation values performed at the end of each generation, showed the emergence of weakly correlated subpopulations, however they disappeared after a large number of generations in favour of a single strongly correlated population at the end of the seed.

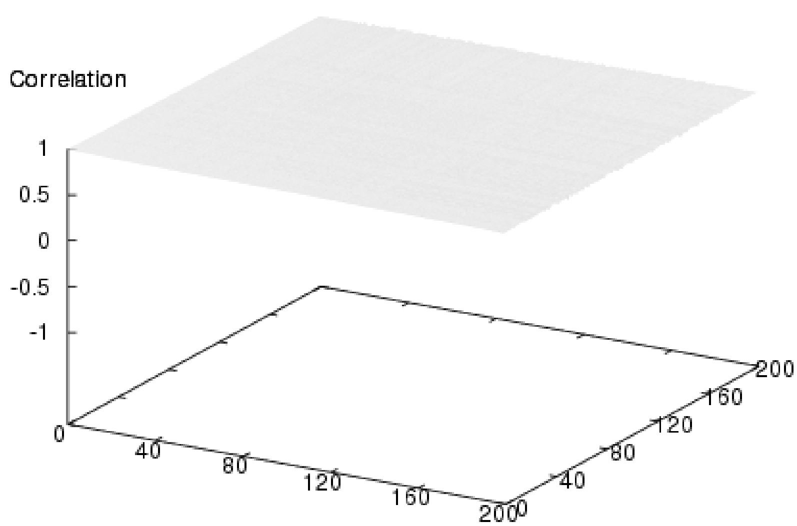

Figure 6: Typical graph of correlation in the simulation with one population and teams changed every epoch

A very interesting situation is that of the second algorithm. Here a single robot did not change team during a generation, as consequence it had the possibility to receive a good value of fitness when it matched a good partner. Then the emergence of innate specialisation was not hindered by the algorithm: basically it depended on the kind of task to be performed. In the light-approaching task, where the goal can be reached without specialisation, at the end of each seed we observed a population without specialised individuals. On the other hand, in the coordinated-motion task, where breaking initial symmetric configurations could result in a considerable advantage, the innate specialisation clearly emerged in every seed. These observations have been corroborated by the correlation analysis made on the population. In Figure 7, the result of a correlation measure after a seed is shown: it is clear that two different genotypes characterized the whole population. By repeating the correlation measure at the end of each generation, we first noticed the emergence of two different subpopulations and then the stabilization of this configuration.

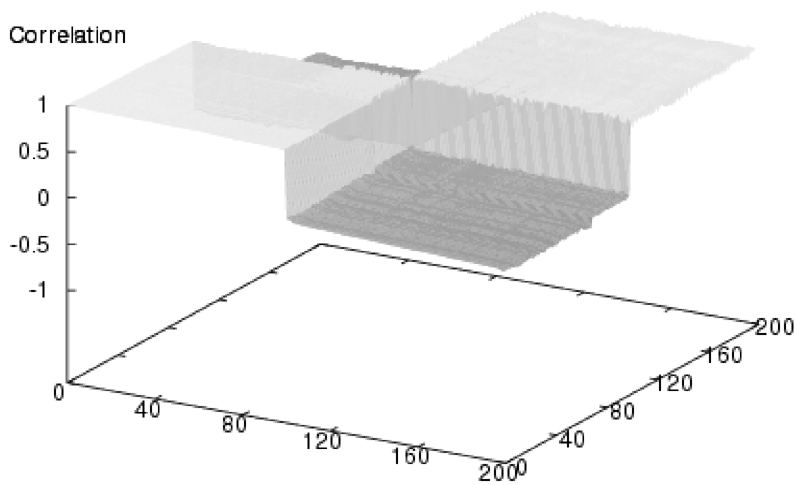

Figure 7: Typical graph of correlation in the simulation with one population and teams fixed for the entire generation.

4.3 Selection at robot level – two populations

In the above cases, the algorithms did not foster the specialisation, but in literature we can find examples of algorithms which could facilitate the emergence of innate specialisation. In [7], the author forced the differentiation of the individuals by using separated populations for each role. Following this experience, we performed some simulations adopting the same strategy. In these new simulations, the population of robots had been divided in two separated sub-populations of 100 individuals each. When a team had to be created, the genetic algorithm randomly drew one individual from each population. Then, as above, we performed two series of tests: in the first, new teams were created at the beginning of each epoch, while in the second teams were created at the beginning of each generation and then remained fixed for the entire generation. As expected, and in accordance with the results obtained in [7], the innate specialisation emerged in all the tests without affecting the goodness of the fitness values. A more accurate analysis performed by the correlation techniques revealed a strong correlation among the genotypes of the same population, whereas, given two individuals belonging to different populations, their correlation was close to zero.

Bibliography

[1] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, “Cooperative mobile robotics: Antecedents and directions”, Autonomous Robots, vol. 4, pp. 1–23, 1997.

[2] G. Dudek, M. Jenkin, and E. Milios, “A taxonomy of multirobot systems”, in Robot Teams: From Diversity to Polymorphism, T. Balch and L. E. Parker, Eds. A K Peters Ltd., Wellesley, MA, 2002.

[3] S. Nolfi and D. Floreano, “Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines”, MIT Press/Bradford Books, Cambridge, MA, 2000.

[4] M. Dorigo, V. Trianni, E. Sahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L. M. Gambardella, “Evolving selforganizing behaviors for a swarm-bot”, Autonomous Robots, vol. 17, no. 2-3, pp. 223-245, 2004.

[5] M. Quinn, L. Smith, G. Mayley, P. Husbands, “Evolving teamwork and role allocation with real robots”, In R.K. Standish, M.A. Bedau, H.A. Abbass (Eds.), Proceedings of the 8th International Conference on Artificial Life, pp. 302-311, The MIT Press, 2002.

[6] G. Baldassarre, S. Nolfi, D. Parisi, “Evolving mobile robots able to display collective behaviour”, Artificial Life, Vol. 9, pp. 255-267, 2003.

[7] F. Gomez, R. Mikkulainen, “2-D Pole balancing with recurrent evolutionary networks”, In Proceedings of the International Conference on Artificial Neural Networks, ICANN-98, Skövde, Sweden, 425-430, Berlin, New York: Springer, 1998.

[8] F. Mondada, E. Franzi, P. Ienne, “Mobile robot miniaturization: a tool for investigation in control algorithms”, In T. Yoshikawa & F. Miyazaki (eds.), Proceedings of the Third International Symposium on Experimental Robotics, Berlin: Springer Verlag, 1993.

[9] O. Miglino, H. H. Lund, and S. Nolfi, “Evolving mobile robots in simulated and real environments”, Artificial Life, vol. 4, pp. 417-434, 1995.

[10] S. Nolfi, “Evorobot 1.1 User Manual”, Rapporto tecnico. Roma: Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerche, Italia, 2000.

[11] T. Ferrauto, G. Baldassarre, G. Di Stefano, D. Parisi, “L'Emergere della specializzazione in squadre di robot”, Proceedings del 2° Workshop Italiano di Vita Artificiale, Roma, 2005.

[12] M. Quinn, “Evolving Communication without Dedicated Communication Channels”, In J. Kelemen and P. Sosìk (eds.), ECAL01, pages 357-366, Prague: Springer, 2001.

[13] P. Stone, M. Veloso, “Task decomposition and dynamic role assignment for real-time strategic teamwork”, in J. Müller, M.P. Singh, A.S. Rao (eds.), Proceedings of the 5th International Workshop on Intelligent Agents: Agent Theories, Architectures, and Languages (ATAL-98), Berlin: Springer-Verlag, 1555, 293-308, 1999.

[14] B.P. Gerkey, M.J. Mataric, “A formal analysis and taxonomy of task allocation in multi-robot systems”, The International Journal of Robotics Research, number 23, pages 939-954, 2004.

[15] Balch, T., “Taxonomies of multirobot task and reward”, Techreport, Robotic Institute, Carnegie Mellon University, Pittsburg, PA, 1998