From Solitary to Collective Behaviours: Decision Making and Cooperation

Vito Trianni[1], Christos Ampatzis[2], Anders Lyhne Christensen[2], Elio Tuci[2], Marco Dorigo[2], and Stefano Nolfi[1]

[1] LARAL, ISTC, CNR, Italy {vito.trianni,stefano.nolfi}@istc.cnr.it
[2] IRIDIA, CoDE, ULB, Belgium {campatzi,alyhne,etuci,mdorigo}@ulb.ac.be

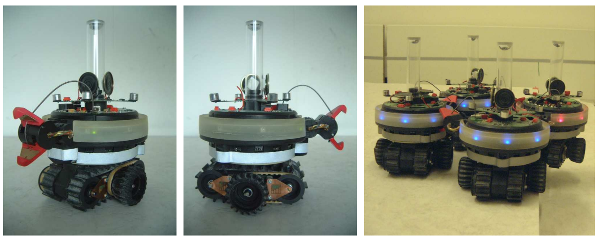
Abstract. In a social scenario, establishing whether a collaboration is required to achieve a certain goal is a complex problem that requires decision making capabilities and coordination among the members of the group. Depending on the environmental contingencies, solitary actions may result more efficient than collective ones and vice versa. In robotics, it may be difficult to estimate the utility of engaging in collaboration versus remaining solitary, especially if the robots have only limited knowledge about the environment. In this paper, we use artificial evolution to synthesise neural controllers that let a homogeneous group of robots decide when to switch from solitary to collective actions based on the information gathered through time. However, being in a social scenario, the decision taken by a robot can influence—and is influenced itself—by the status of the other robots that are taking their own decisions at the same time. We show that the simultaneous presence of robots trying to decide whether to engage in a collective action or not can lead to cooperation in the decision making process itself.

1 Introduction

Decision making is a complex problem for a collective robotic system, due to the necessity to reach a global consensus among the robots, which contrasts with the system's inherent decentralisation. Current approaches resort to biological inspiration [1–3] or to context-specific solutions [4, 5]. The problem of deciding whether to switch between solitary and collective behaviours is much less studied. Such a problem is of fundamental importance for a particular robotic system: the *swarm-bot*, a swarm robotic artefact composed of a number of autonomous mobile robots—referred to as *s-bots*—which have the ability to connect to each other forming a physical structure, as shown in Fig. 1 (for more details, see [6]). Forming a *swarm-bot* by self-assembly is a collective action that can lead to great advantages: for example, the *swarm-bot* can overcome an obstacle impassable for solitary *s-bots* [7] or collectively transport a heavy item [8]. On the other hand, for tasks such as searching for a goal location or tracing an optimal path to a goal, a swarm of unconnected *s-bots* may be more efficient [9].

When should a group of *s-bots* assemble in a *swarm-bot*? This problem—referred to as *functional self-assembly* [10]—has been studied to date without particular focus on the decision making process that should lead to the switch from individual to collective behaviours. The decision to self-assemble was based either on *a priori* assumptions or on clearly distinguishable environmental cues [8, 7, 10], which may reduce the adaptiveness of a solution and the efficiency of the system as a whole. We believe that a truly adaptive system should prove capable of autonomously extracting all the information it requires to solve a problem. In other words, the *s-bots* should be capable of recognising the necessity to self-assemble based only on the environmental contingencies they experience. Given the limited sensory range of each *s-bot*, the information relevant to autonomously decide whether to switch from a solitary to a collective behaviour is not ready-to-use, but should be constructed by the robots while they interact and accumulate experience about the environment in which they are placed. Moreover, being in a collective scenario, the actions of each *s-bot* can influence—and are influenced themselves—by the status of the other *s-bots*, which try to make their own decisions at the same time. This opens the way to cooperative solutions that can exploit not only the dynamical interactions among individuals, but also the way in which these interactions change over time. In this paper, we show how the adaptiveness of the robots' behaviour can be increased by an evolutionary process that favours through selection those solutions that improve the "fitness" of the robotic group. Here, we do not focus on assembly but we limit our study to the processes that should lead to the formation of a *swarm-bot*. We demonstrate how non-trivial individual and collective decision making processes can be efficiently obtained.

The work presented in this paper is based on previous studies about time-dependent decision making mechanisms [11, 12]: robots had to categorise the environment in which they were placed, either by explicit signalling or by performing different actions in different environments. When a social scenario was considered, communication resulted in increased robustness of the categorisation [12]. In this paper, we advance by studying a collective behaviour—i.e., aggregation—as a consequence of the decision making process: robots are placed in two different environments and, according to the environmental contingencies they experience, they should perform the appropriate individual or collective action. From the observer—i.e., *distal*—point of view, this is yet another categorisation problem in which the robotic group faces a binary choice between two



(a) (b) (c)

Fig. 1. (a,b) Different views of an *s-bot*. (c) A *swarm-bot* in a indoor environment.

environment types. However, from the robot—i.e., *proximal*—point of view, the binary choice is to be performed between two different behavioural states: a solitary behaviour and a collective one. In the definition of the evaluation function, we emphasise the importance of evaluating the robots for their ability to switch between behavioural states (see Section 2.3). The obtained results show that a number of different strategies can be evolved to solve the given problem. Among these, we show that those solutions that exploit communication perform better, systematically achieving a consensus in the group and reducing the decision errors.

2 The Task

The path towards the evolution of neural controllers for functional self-assembly in a physical *swarm-bot* passes through the definition of the following experimental scenario. A group of *s-bots* is placed in an arena that is surrounded by some obstacles that *s-bots* cannot overcome individually. The arena may have a *way out*, that is, a passage through which a solitary *s-bot* can exit (see Figure 2a). However, an *s-bot* does not have the perceptual abilities to detect the *way out* from every location in the arena. Therefore, *s-bots* should first search for the *way out* and, if they do not find any as in Figure 2b, they should aggregate and self-assemble in order to collectively overcome the obstacles that surrounds the arena. As mentioned above, we consider in this paper only the first part of this scenario concerning the decision to switch from the individual behaviour of searching for the *way out* to the collective behaviour of aggregating in one place. The second part of the scenario concerning self-assembly is on-going work.

2.1 The *S-bot*

An *s-bot* is a small mobile autonomous robot with self-assembling capabilities, shown in Fig. 1a and b [6]. The main body is a cylindrical turret with a diameter of about 12 cm. The turret holds the gripper used for assembling with other *s-bots* and can be actively rotated with respect to the chassis. The traction system is composed of both tracks and wheels, and provides a differential drive motion.[3] Each *s-bot* is provided with many sensory systems, useful for the perception of the surrounding environment or for proprioception. In this paper, we make use of the four proximity sensors placed under the chassis—referred to as *ground sensors*—that can be used for perceiving the ground's grey level. Each robot is also equipped with an omni-directional camera and red LEDs distributed around the *s-bots*' turret. The circular image recorded by the camera is filtered in order to return the distance of the closest *s-bot* in each of four 90° sectors, up to a maximum distance of about 50 cm. In order to communicate with each other, *s-bots* are provided with a very simple sound signalling system, which can produce

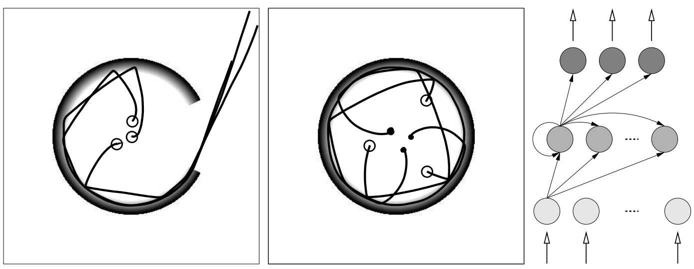
[3] The experiments presented here are performed in simulation only. However, we already presented elsewhere the portability of similar controllers to the physical system [12].

a continuous tone with fixed frequency and intensity. When a tone is emitted, it is perceived by every robot in the arena, including the signalling *s-bot*. The tone is perceived in a binary way, that is, either there is someone signalling in the arena or there is no one.

2.2 Experimental Setup

Three *s-bots* are initially placed up to 25 cm from the centre of a boundless arena. The arena contains a circular band in shades of grey (inner radius: 1.0 m; outer radius: 1.2 m—see Fig. 2a,b). The outer border of the circular band is painted in black and simulates the presence of a trough/obstacle that the *s-bots* cannot overcome individually: the simulation is stopped whenever individual *s-bots* pass over the black border, and the trial is considered unsuccessful. The grey level of the circular band can be perceived by the *s-bots* only locally through the ground sensors, and it is meant to warn *s-bots* about the presence of the simulated trough/obstacle: the darker the ground colour, the closer the danger. The *s-bots* can be placed in two different environments: in environment A, the circular band is discontinuous—i.e., there is a *way out* through which the *s-bots* can exit (see the trajectories in Fig. 2a). In environment B, the *way out* is not present and therefore *s-bots* should aggregate after having searched for it (see the trajectories in Fig. 2b). The amplitude of the *way out* is randomly selected in each trial within the interval $[\pi/4, \pi/2]$.

Homogeneous groups of *s-bots* are controlled by artificial neural networks, whose parameters are set by an evolutionary algorithm. A single genotype is used to create a group of individuals with an identical control structure. Each *s-bot* is controlled by a continuous time recurrent neural network (CTRNN, see [13]) with a multi-layer topology, as shown in Fig. 2c. The neural network is composed of 9 input neurons ($N_{I,i}$) which are simple relay units, 3 output neurons ($N_{O,i}$) with a sigmoid transfer function, and 5 continuous time hidden



(a) (b) (c)

Fig. 2. (a,b) The experimental arena contains a circular band in shades of grey, which may or may not have the *way out*. Dark lines represent the trajectory of three *s-bots*, and the starting position on the trajectories is indicated by empty circles. (c) The three-layer architecture of the neural controller. The hidden layer is composed of continuous time neurons with fully recurrent connections.

neuron ($N_{H,i}$). Four inputs take values from the camera, four from the ground sensors and one from sound perception, while two outputs control the wheels and one controls the sound signal. The weights of the synaptic connections between neurons, the bias terms and the decay constants of the hidden neurons are genetically encoded parameters, optimised through a simple generational evolutionary algorithm [14]. The population contains 100 genotypes that are evolved for 5000 generations. Each genotype is a vector of 98 real values (85 synaptic connections, 5 decay constants and 8 bias terms). Subsequent generations are produced by a combination of selection with elitism and mutation. Recombination is not used. At every generation, the best 20 genotypes are selected for reproduction, and each generates 4 offspring. The genotype of the selected parents is copied in the subsequent generation; the genotype of the 4 offspring is mutated with a 50% probability of adding a random Gaussian offset to each real-valued gene.[4]

2.3 The Evaluation Function

During evolution, a genotype is mapped into a control structure that is cloned and downloaded onto all the *s-bots* taking part in the experiment. The fitness of a genotype is the average performance of a group of three *s-bots* evaluated over ten trials—five performed in environment A and five in environment B.[4] Each trial lasts 65 seconds and differs from the others in the initialisation of the random number generator, which influences mainly the *s-bots* starting positions and orientations, and the amplitude of the *way out*, if present. As mentioned above, robots should make a binary choice between two behavioural states: (i) searching for the *way out* and moving away from the arena centre—hereafter called solitary state $\mathcal{S}$—or (ii) aggregating with the other *s-bots*—hereafter called collective state $\mathcal{C}$. The performance of the group is computed as the average individual performance of the three *s-bots*. The individual performance rewards the movements of an *s-bot* according to its current behavioural state. When in state $\mathcal{S}$, the *s-bot* should continue to move away from the centre, and it is considered successful if it reaches the distance $D_O(\mathcal{S}) = 2.4$ m from the centre. When an *s-bot* switches to state $\mathcal{C}$, it should aggregate with the other robots by reducing its distance from the centre of mass of the group. It is considered successful if it stays below the distance $D_O(\mathcal{C}) = 0.25$ m from the centre of mass of the group. In both cases, we conventionally say that a successful *s-bot* "achieves the desired distance D_O". Note that a trial is terminated whenever an *s-bot* passes over the black border of the circular band—and in this case its performance is 0—or if *s-bots* collide when in state $\mathcal{S}$. It is worth mentioning that when computing the individual performance, the behavioural state of an *s-bot* cannot be directly observed, because it is not explicitly encoded in the controller or elsewhere. However, knowing the environment type and looking at the movements of the robot, it is possible to estimate in which state an *s-bot* should be at any given time: when an *s-bot* is placed in environment A, it should search for the *way out* and exit through it, therefore it should be in state $\mathcal{S}$.

[4] For more details, see the supplementary material available in [15].

When an *s-bot* is placed in environment B, it should initially search for the *way out*, being in state $\mathcal{S}$, and at some point it should give up and aggregate, therefore switching to state $\mathcal{C}$. Given that it is not possible to exactly recognise when an *s-bot* switches to state $\mathcal{C}$, we compute the individual performance by considering an *s-bot* in state $\mathcal{C}$ as soon as it encounters the circular band for the first time. On the basis of such estimation of the behavioural state, it is possible to systematically evaluate the *s-bot*'s performance. Note that the evaluation function does not explicitly reward either cooperation or communication. It rather rewards those agents that perform the correct movements in each behavioural state, without any reference to the mechanism necessary to switch from one state to the other.

3 Results

We performed 20 replications of the experiment, most of which were successful. For each evolutionary run, we selected a single controller from the last generation. To do so, we evaluated the 20 best individuals—the *elite* of the last generation—for 200 trials in both environments, and we selected the genotype with the highest average performance. As a result, we obtained 20 controllers—hereafter referred to as $C_1, \ldots, C_{20}$—that were further evaluated for 2000 trials, half in environment A and half in environment B. The obtained results are summarised in Table 1: in both environments, we computed the average performance and its standard deviation (avg $\pm$ std), the rates of success %S (all *s-bots* achieve the desired distance D_O), failure %F (no *s-bot* achieves the desired distance D_O), partial success/failure %M (not all *s-bots* are successful or fail) and error %E (*s-bots* collide or cross the black edge of the circular band). In each trial, we also computed the *coverage*, which is defined as the percentage of the circular band that each robot covers in average during a trial: a value smaller than 1 indicates that the single *s-bot* does not search the whole circular band for the *way out*, while a value bigger than 1 indicates that the single *s-bot* performs more than one tour (see Fig. 3). The coverage—together with the success rate—is useful to quantitatively assess the quality of the evolved strategies.

Successful controllers produce good search behaviours when *s-bots* are in state $\mathcal{S}$:[5] *s-bots* avoid collisions and move away from the centre of the arena. Once on the circular band, *s-bots* start looping in search of the *way out*, which is eventually found and traversed when *s-bots* are placed in environment A. On the contrary, if *s-bots* are placed in environment B, the absence of the *way out* is recognised by the *s-bots* through the integration over time of their perceptual flow, which includes the signals that the *s-bots* may emit (for more insights about decision making processes based on temporal cues, see [11, 12]). As a consequence, a behavioural transition from state $\mathcal{S}$ to state $\mathcal{C}$ can be observed. The modalities with which the transition is performed significantly vary across the different solutions synthesised during different evolutionary runs. However, looking at the behaviour produced by the evolved controllers, we recognised some similarities that let us classify the controllers in 4 classes.

[5] Detailed descriptions and movies are available as supplementary material in [15].

Table 1. Post-evaluation results. See text for details.

		environment A					environment B				
		avg $\pm$ std	%S	%M	%F	%E	avg $\pm$ std	%S	%M	%F	%E
U	c_4	0.82 ± 0.14	92.0	6.5	1.0	0.5	0.37 ± 0.11	19.4	18.9	61.7	0.0
	c_6	0.85 ± 0.06	98.6	1.2	0.0	0.2	0.31 ± 0.08	0.9	30.6	68.4	0.1
	c_{14}	0.83 ± 0.15	91.3	6.2	0.0	2.5	0.46 ± 0.15	2.5	65.1	24.0	8.4
	c_{17}	0.66 ± 0.07	74.3	25.4	0.1	0.2	0.39 ± 0.08	4.9	78.8	16.3	0.0
B	c_1	0.86 ± 0.11	97.7	0.8	0.0	1.5	0.69 ± 0.07	95.9	2.8	1.3	0.0
	c_5	0.85 ± 0.13	92.1	5.7	0.0	2.2	0.57 ± 0.14	66.8	16.9	16.1	0.2
	c_8	0.83 ± 0.15	90.3	7.6	0.4	1.7	0.57 ± 0.12	34.3	55.2	9.2	1.3
	c_{10}	0.88 ± 0.07	99.0	0.6	0.0	0.4	0.66 ± 0.07	94.1	2.1	3.7	0.1
	c_{16}	0.85 ± 0.14	94.4	4.1	0.0	1.5	0.74 ± 0.13	94.1	2.3	1.4	2.2
M	c_3	0.83 ± 0.15	85.8	11.7	0.0	2.5	0.63 ± 0.09	87.6	8.1	3.4	0.9
	c_7	0.79 ± 0.20	89.3	5.5	0.0	5.2	0.62 ± 0.25	49.5	34.2	10.5	5.8
	c_{11}	0.86 ± 0.07	98.9	0.6	0.0	0.5	0.61 ± 0.07	87.6	9.5	2.7	0.2
	c_{13}	0.85 ± 0.09	94.3	5.2	0.0	0.5	0.62 ± 0.07	93.0	5.3	0.8	0.9
	c_{19}	0.81 ± 0.15	94.8	2.3	0.6	2.3	0.67 ± 0.12	91.7	3.8	1.9	2.6
	c_{20}	0.87 ± 0.06	99.6	0.0	0.0	0.4	0.59 ± 0.07	79.3	11.3	9.3	0.1
C	c_2	0.86 ± 0.10	98.6	0.1	0.0	1.3	0.82 ± 0.12	97.1	0.4	0.9	1.6
	c_9	0.87 ± 0.08	99.2	0.0	0.0	0.8	0.78 ± 0.12	88.1	8.3	3.1	0.5
	c_{12}	0.87 ± 0.05	99.6	0.3	0.0	0.1	0.74 ± 0.11	87.8	6.4	5.4	0.4
	c_{15}	0.86 ± 0.08	99.3	0.0	0.0	0.7	0.78 ± 0.13	96.6	0.4	0.6	2.4
	c_{18}	0.84 ± 0.18	95.8	0.0	0.0	4.2	0.83 ± 0.17	95.3	0.3	1.0	3.4

Class $\mathbf{U} = \{C_4, C_6, C_{14}, C_{17}\}$ encompasses the "unsuccessful" controllers, that is, those controllers that solve the task only in part. These controllers generally produce appropriate search behaviours when *s-bots* are in state $\mathcal{S}$, as confirmed by the good performance and the high success rate in environment A (see Table 1). However, when *s-bots* are placed in environment B they fail in systematically aggregating, scoring a low performance and a poor success rate. The second class $\mathbf{B} = \{C_1, C_5, C_8, C_{10}, C_{16}\}$ consists of controllers that produce a strategy named "bouncing" after the aggregation behaviour of the *s-bots* in state $\mathcal{C}$: *s-bots* search for each other by continuously bouncing off the circular band, so that they sooner or later meet and remain close. Communication is not exploited,[6] and consequently each *s-bot* individually switches from state $\mathcal{S}$ to state $\mathcal{C}$, without any reference to the state of the other robots. The bouncing behaviour is resilient to possible individual failures in environment A: by bouncing off the circular band, *s-bots* can continue searching for the *way out*, even if less efficiently. This corresponds to high success rates in environment A despite the fact that the *s-bots* perform in average less than one tour over the circular band, as indicated by the corresponding coverage (see Fig. 3). The third class $\mathbf{M} = \{C_3, C_7, C_{11}, C_{13}, C_{19}, C_{20}\}$ encompasses controllers that produce a strategy named "meeting", due to the fact that *s-bots* aggregate by encountering at a meeting point, which is normally close to the centre of the arena. Except for C_7

[6] Only C_{16} exploits signalling to trigger a synchronous switch to state $\mathcal{C}$ [15].

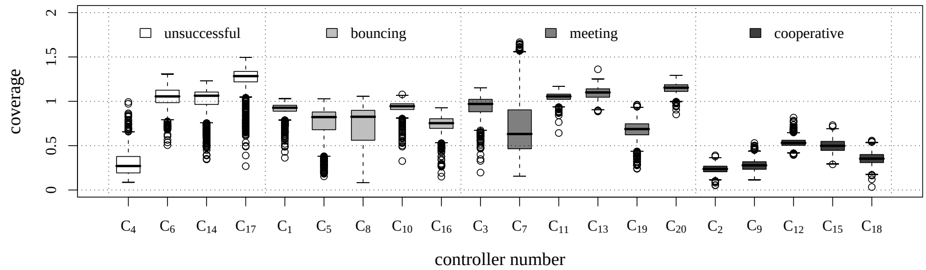


Fig. 3. The *coverage* of the evolved controllers. Boxes represent the inter-quartile range of the data, while the horizontal lines inside the boxes mark the median values. The whiskers extend to the most extreme data points within 1.5 times the inter-quartile range from the box. The empty circles mark the outliers.

and C_{19}, controllers of this class do not make use of communication. The main difference with class **B** controllers resides in the aggregation behaviour, which lets robots leave the band and move in circles close to the centre of the arena, waiting for the other *s-bots* to reach a similar position. This behaviour is not robust with respect to possible decision errors in environment A. As a consequence, evolution shaped the controllers of this class to be characterised by a high coverage (see Fig. 3): *s-bots* perform more than one loop over the circular band before switching to state $\mathcal{C}$, which corresponds to robust individual decisions and a high success rate in environment A. The last class $\mathbf{C} = \{C_2, C_9, C_{12}, C_{15}, C_{18}\}$ is named "cooperative" because it encompasses controllers that produce communicative behaviours exploited for cooperation in the decision making. In fact, *s-bots* are able to share the information they collect over time through their signalling behaviour. The *s-bots* initially emit a sound signal, and they stop only after looping on the circular band for some time. If any robot finds the *way out*, signalling continues, inducing all other *s-bots* to remain in state $\mathcal{S}$ and to keep searching for the *way out*. This leads to a high success rate in environment A, and no complete failures are observed (see Table 1). When the *way out* is not present, all robots eventually stop signalling, allowing the transition to state $\mathcal{C}$ and triggering the aggregation behaviour. By sharing the information through communication, *s-bots* can collectively search the circular band, splitting the task among them: as shown by the coverage data in Fig. 3, each *s-bot* covers from a quarter to half circle when placed in environment B. This allows to consistently reduce the search time, achieving high performance and high success rates. Communication is fundamental here, because it provides robustness to the decision making process and it makes the system more efficient by reducing the time necessary to take the decisions to switch from solitary to collective behaviours.

In order to quantitatively compare the performance of the behaviours produced by the evolved controllers, we used the performance data recorded over 2000 trials to perform a series of pairwise Wilcoxon tests among all possible controller couples, which allowed to produce the following ranking:

$$C_4 \prec C_6 \prec C_{17} \prec C_{14} \prec C_3 \prec C_8 \prec \{C_{13}, C_{11}\} \prec C_{19} \prec C_1 \prec$$
$$\prec C_{20} \prec C_{10} \prec C_5 \prec C_7 \prec \{C_{16}, C_{12}\} \prec C_{15} \prec C_9 \prec C_2 \prec C_{18},$$

where $C_i \prec C_j$ indicates that C_j is statistically better than C_i with 99% confidence. Controllers that have no statistical difference are reported in curly brackets. All class **U** controllers have a low rank, as one would expect. Instead, it is worth noting that class **C** controllers perform statistically better than the others. Moreover, other controllers making use of communication but with a different strategy (namely C_7-Meeting and C_{16}-Bouncing) occupy a good position in the rank. We can conclude that communication can improve the efficiency and the robustness of the decision making process. Robots exploiting only local interactions are prone to decision errors or to behaviours that are less efficient. Therefore, by cooperating through communication, *s-bots* increase their ability to make correct and unanimous decisions, consequently achieving a better performance.

4 Conclusions

We have studied the decision making mechanisms that can let a group of robots switch from solitary to collective behaviours. We have faced the problem through an evolutionary approach in order to limit the *a priori* assumptions and search broadly the space of the possible solutions. The results we obtained demonstrate that suitable decision making mechanisms can be evolved. Moreover, by providing the robots with a simple communication channel, the evolved cooperative strategies display higher efficiency and enhanced robustness of the system. The use of communication generally results in a faster and more robust decision making process. Communication increases the otherwise limited information available to each robot, not only about the quality of the physical environment but also and above all about the social environment and about the internal states of other robots that, by definition, are not directly accessible.

A systematic analysis of the evolutionary pressures that shaped the above mechanisms is out of the scope of this paper, and is left for future work. Further testing with real robots is also planned for the future. Finally, we plan to integrate the decision making processes studied here with on-going work on self-assembly, in order to produce the first example of functional self-assembly of real *swarm-bots* based on completely evolved controllers.

Acknowledgements

This work was supported by the Swarmanoid and by the ECAgents projects. Both projects are funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission, under grant IST-022888 and 001940 respectively. The authors thank Paola Pellegrini and Mauro Birattari for providing the tools necessary to perform the statistical analysis.

References

1. Parker, C.A.C., Zhang, H.: Biologically inspired decision making for collective robotic systems. In: Proc. of the 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IEEE Comp. Soc. Press, Los Alamitos, CA (2004) 375–380
2. Parker, C.A.C., Zhang, H.: Active versus passive expression of preference in the control of multiple-robot decision-making. In: Proc. of the 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IEEE Comp. Soc. Press, Los Alamitos, CA (2005) 3706–3711
3. Garnier, S., Jost, C., Jeanson, R., Gautrais, J., Asadpour, M., Caprari, G., Theraulaz, G.: Aggregation behaviour as a source of collective decision in a group of cockroach-like robots. In Capcarrere, M.S., *et al.*, eds.: Advances in Artificial Life. Proc. of the 8th European Conference on Artificial Life. Volume 3630 of LNCS., Springer Verlag, Berlin, Germany (2005) 169–178
4. Kok, J.R., Spaan, M.T.J., Vlassis, N.: Multi-robot decision making using coordination graphs. In: Proc. of the 11th Int. Conf. on Advanced Robotics, IEEE Press, Piscataway, NJ (2003) 953–957
5. Vlassis, N., Elhorst, R., Kok, J.R.: Anytime algorithms for multiagent decision making using coordination graphs. In: Proc. of the 2004 IEEE Conf. on System, Man and Cybernetics, IEEE Press, Piscataway, NJ (2004) 953–957
6. Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I.V., Floreano, D., Deneubourg, J.L., Nolfi, S., Gambardella, L.M., Dorigo, M.: SWARM-BOT: A new distributed robotic concept. Auton. Robots **17**(2–3) (2004) 193–221
7. O'Grady, R., Groß, R., Bonani, M., Mondada, F., Dorigo, M.: Self-assembly on demand in a group of physical autonomous mobile robots navigating rough terrain. In Capcarrere, M.S., *et al.*, eds.: Advances in Artificial Life. Proc. of the 8th Europ. Conf. on Artificial Life. Volume 3630 of LNCS., Springer Verlag, Berlin, Germany (2005) 272–281
8. Groß, R., Tuci, E., Bonani, M., Mondada, F., Dorigo, M.: Object transport by modular robots that self-assemble. In: Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation, IEEE Comp. Soc. Press, Los Alamitos, CA (2006) 2558–2564
9. Nouyan, S., Dorigo, M.: Chain based path formation in swarms of robots. In Dorigo, M., *et al.*, eds.: Ant Colony Optimization and Swarm Intelligence: 5th Int. Workshop, ANTS 2006. Volume 4150 of LNCS., Springer Verlag, Berlin, Germany (2006) 120–131
10. Trianni, V., Tuci, E., Dorigo, M.: Evolving functional self-assembling in a swarm of autonomous robots. In Schaal, S., *et al.*, eds.: From Animals to Animats 8. Proc. of the 8th Int. Conf. on Simulation of Adaptive Behavior, MIT Press, Cambridge, MA (2004) 405–414
11. Tuci, E., Trianni, V., Dorigo, M.: 'Feeling' the flow of time through sensorimotor co-ordination. Conn. Sciene **16**(4) (2004) 301–324
12. Ampatzis, C., Tuci, E., Trianni, V., Dorigo, M.: Evolution of signalling in a swarm of robots controlled by dynamic neural networks. In Şahin, E., *et al.*, eds.: Swarm Robotics. Volume 4433 of LNCS., Springer Verlag, Berlin, Germany (2007) 173–188
13. Beer, R.D.: A dynamical systems perspective on agent-environment interaction. Art. Intell. **72** (1995) 173–215
14. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA (1989)
15. Trianni, V., Ampatzis, C., Christensen, A.L., Tuci, E., Dorigo, M., Nolfi, S.: From solitary to collective behaviours: Decision making and cooperation – supplementary material. `http://laral.istc.cnr.it/esm/trianni-etal-ecal2007.html`