Self-assembly on Demand in a Group of Physical Autonomous Mobile Robots Navigating Rough Terrain

Rehan O'Grady[1], Roderich Groß[1], Francesco Mondada[2], Michael Bonani[2], and Marco Dorigo[1]

[1] IRIDIA, Université Libre de Bruxelles, Brussels, Belgium
{rogrady, rgross, mdorigo}@ulb.ac.be
[2] ASL, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
{francesco.mondada, michael.bonani}@epfl.ch

Abstract. Consider a group of autonomous, mobile robots with the ability to physically connect to one another (self-assemble). The group is said to exhibit *functional self-assembly* if the robots can choose to self-assemble in response to the demands of their task and environment [15]. We present the first robotic controller capable of functional self-assembly implemented on a real robotic platform.

The task we consider requires a group of robots to navigate over an area of unknown terrain towards a target light source. If possible, the robots should navigate to the target independently. If, however, the terrain proves too difficult for a single robot, the robots should self-assemble into a larger group entity and collectively navigate to the target.

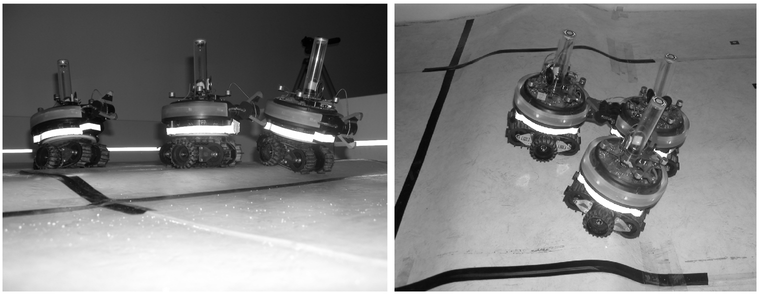
We believe this to be one of the most complex tasks carried out to date by a team of physical autonomous robots. We present quantitative results confirming the efficacy of our controller. This puts our robotic system at the cutting edge of autonomous mobile multi-robot research.

1 Introduction

Collective robotics addresses the design, implementation and study of multi-robotic systems. Swarm robotics is a subset of collective robotics which takes inspiration from social insect behaviour and emphasises *swarm intelligence* [2] principles such as decentralisation of control and use of local information. Many swarm robotics applications require cooperation between robots [8]. Some applications further require physical connectivity between cooperating robots. It is this last class of application that interests us. Although there is a large body of work on the capabilities of physically connected systems, very little research has been conducted on the mechanisms of when and how autonomous mobile agents should self-assemble.

The phrase *functional self-assembly* [15] describes a key adaptive response mechanism of distributed systems. We define self-assembly as the process

M. Capcarrere et al. (Eds.): ECAL 2005, LNAI 3630, pp. 272–281, 2005.



(a) (b)

Fig. 1. (a) *S-bots* overcome a 2 cm hill independently. (b) *S-bots* self-assemble in order to overcome a 5 cm hill collectively.

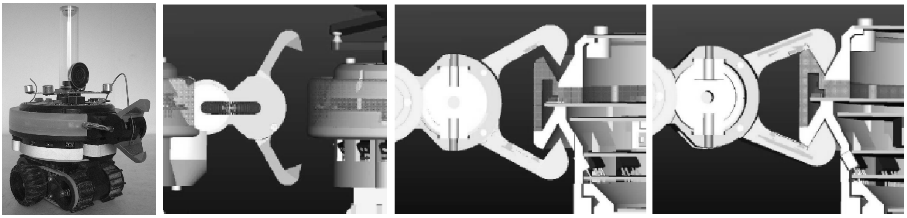
through which separate autonomous agents form a larger group entity by physically connecting to one another. If the agents can autonomously choose to self-assemble in response to the demands of their task and environment, they are said to display *functional self-assembly.*

A number of social insect species depend on functional self-assembly (for a review see [1]). Members of the ant species *Œcophylla longinoda*, for example, connect to one another to form bridges that other ants can then traverse [7]. Given its ubiquity in natural systems, functional self-assembly has been given surprisingly little attention by the swarm robotics community. In the only dedicated work, Trianni et al. [15] evolved neural network controllers for robots that needed to self-assemble and disassemble in order to traverse artificially designated 'hot' and 'cold' zones in a simple simulation environment.

Over the last decade, much of the research involving systems of physically connected robotic modules has been targeted at collective rough terrain navigation. In Hirose *et al.*'s system [6] modules are mechanically linked by means of a passive arm and are therefore incapable of self-assembly. Yim *et al.*'s system [16] can climb near vertical walls. Individual modules are incapable of autonomous motion and have very few external sensors for perception of the environment. Similar limitations are found in the majority of self-reconfigurable robotic systems, usually rendering self-assembly difficult or impossible [12,14].

In this paper we present the first physical robot controller capable of functional self-assembly. Our controller was implemented on the SWARM-BOT robotic platform [11,10,3]. This innovative system consists of a number of autonomous robotic agents called *s-bots*. *S-bots* are able to physically connect to one another, thus forming a larger group entity termed a *swarm-bot*. A *swarm-bot* can complete tasks impossible for a single *s-bot*. It can, for example, cross chasms wider than an *s-bot* or overcome hills too steep for a single *s-bot*.

The task we investigate requires a group of *s-bots* to navigate towards a target light source over unknown terrain. The *s-bots* must 'decide' whether or not to self-assemble based on the terrain they encounter. We use two different environments in our experiments. The first environment contains a simple hill which a single *s-bot* can overcome (see Fig. 1a). The *s-bots* can thus reach the



(a) (b)

Fig. 2. (a) The *s-bot*. (b) The *s-bot* gripping mechanism.

target independently. The second environment contains a steep hill too difficult for a single *s-bot*. The *s-bots* must self-assemble in order to overcome the hill and reach the target (see Fig. 1b).

2 Experimental Setup

2.1 The S-Bot

This study was conducted on the SWARM-BOT robotic platform [11,10,3]. The system consists of a number of mobile autonomous robots called *s-bots* (see Fig. 2a). The *s-bot* is equipped with a traction system made up of tracks and wheels. This chassis provides the *s-bot* with efficient on the spot rotation and mobility on moderately rough terrain. The majority of the *s-bot* sensory and processing systems are housed in a turret mounted above the chassis. A motorised axis allows this turret to rotate with respect to the chassis.

Physical connections between two *s-bots* can be established by a gripper-based connection mechanism (see Fig. 2b). Each *s-bot* is surrounded by a T-shaped ring which can be grasped by other *s-bots*.

The *s-bot* sensory systems used in this study are as follows: 15 proximity sensors distributed around the turret allow for the detection of obstacles. A 3-axes accelerometer provides information on the *s-bots*' inclination which can be used to detect if the *s-bot* is in danger of falling. The connection ring of the *s-bot* is equipped with eight groups of coloured LEDs. An omni-directional camera is mounted on the turret. The combination of the camera and the LED ring allows an *s-bot* to communicate its presence and even its internal state to other nearby *s-bots*. Inside the gripper is an optical light barrier to detect the presence of objects to be grasped. Other sensors provide the *s-bot* with information about its internal motors. This includes positional information (e.g., of the rotating turret) and torque information (e.g., of forces acting on the tracks).

2.2 The Task

We conduct experiments in two different environments (see Fig. 3). Both measure 240 cm x 120 cm and consist of two areas of flat terrain (a starting area and a target area) separated by an area of rough terrain. In *Environment A*, the rough

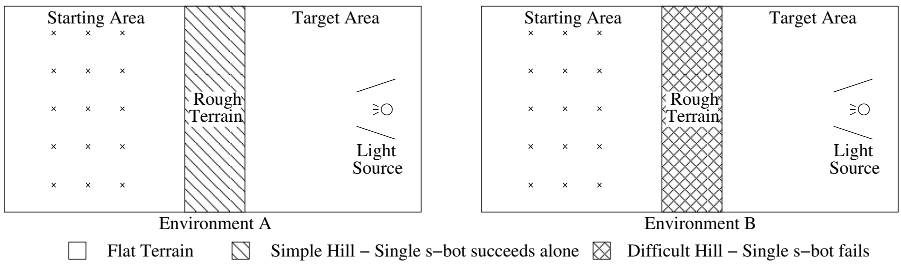


Fig. 3. Scale diagram of the two experimental environments (view from above). *S-bot* starting positions are marked by crosses.

terrain is a 2 cm high hill which can be overcome by a single *s-bot* (see Fig. 1a). In *Environment B* the rough terrain hill is 5 cm high - too difficult for a single *s-bot* (see Fig. 1b).

The initial position of each *s-bot* in the starting area is assigned randomly by uniformly sampling without replacement from a set of 15 possible starting points. The *s-bot*'s initial orientation is chosen randomly from a set of 4 possible directions. To complete the task the *s-bots* must reach the target area without toppling over.

The *s-bots* have no a priori knowledge of the environment they are in—they must react to the environment and determine whether or not to self-assemble. In *Environment A* the *s-bots* should navigate to the target area independently. In *Environment B* the *s-bots* must aggregate, self-assemble and collectively overcome the hill in order to reach the target area.

3 Controller

We use a distributed behaviour-based controller (see Fig. 4). Each *s-bot* is fully autonomous. The same controller is executed on every *s-bot*. An *s-bot* starts by navigating independently towards the target light source. If the *s-bot* finds a hill too difficult for it to pass alone, or if it sees another *s-bot* that is either aggregating or assembled (sees blue or red), it illuminates its blue LEDs and starts aggregating. An aggregating *s-bot* can probabilistically trigger self-assembly by

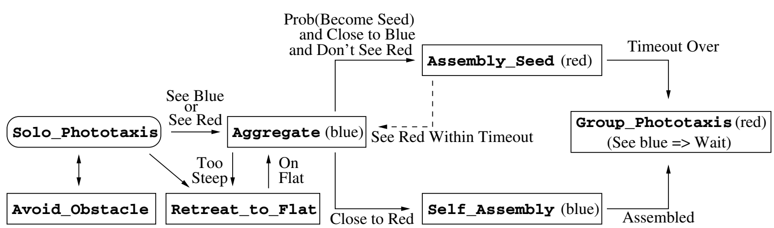


Fig. 4. Behaviour transition model for the behaviour-based controller

Aggregate behaviour

```
loop
  if canSeeClose( red ) then
    switchBehaviour( Self_Assembly )
  else if canSeeFar( red ) then
    approachRed( )
  else if canSeeClose( blue ) then
    Prob(0.04)
      → switchBehaviour( Assembly_Seed )
    Prob(0.96) → doNothing( )
  else if canSeeFar( Blue ) then
    approachBlue( )
  else
    randomWalk( )
  end if
end loop
```

Self_Assembly behaviour

```
repeat
  (i1, i2) ← featureExtraction(camera)
  (i3, i4) ← sensorReadings(proximity)
  (o1, o2, o3) ← f(i1, i2, i3, i4)
  if graspingRequirementsMet(o3) then
    try to grasp
  else
    applyValuesToTracks( o1, o2 )
  end if
until successfully connected
```

Fig. 5. Algorithms for `Aggregate` behaviour (left) and `Self_Assembly` behaviour (right)

illuminating its red LEDs and becoming a static seed. Aggregating *s-bots* assemble to the seed *s-bot* or to already assembled *s-bots* (any red object). Assembled *s-bots* illuminate their red LEDs then perform group phototaxis once they can no longer detect any unassembled *s-bots* (can no longer see blue).

- `Solo_Phototaxis`. This is the starting behaviour. The *s-bot* uses its camera to navigate towards the target light source. The *s-bot* uses its accelerometers to reduce maximum track speed as a linear function of inclination. This is to prevent the *s-bot* toppling before `Retreat_to_Flat` behaviour is triggered.
- `Avoid_Obstacle`. This behaviour is triggered when the readings from the *s-bot*'s 15 proximity sensors exceed a threshold. The *s-bot* determines the direction of the obstacle using its proximity sensors then moves in the opposite direction until the proximity threshold is no longer exceeded.
- `Retreat_to_Flat`. This behaviour is initiated when the *s-bot*'s accelerometers indicate that the *s-bot* is in danger of toppling over. The *s-bot* reverses downhill to flat terrain, reverses away from the rough terrain, then rotates to face away from the slope.
- `Aggregate`. This behaviour is detailed in Fig. 5 (left). The *s-bots* must locate and then approach each other as a precondition for self-assembly. Values for the hard coded probabilities were manually optimised through trial and error.
- `Self_Assembly`. This behaviour is detailed in Fig. 5 (right). Function f maps sensory input (i_1, i_2, i_3, i_4) to motor commands (o_1, o_2, o_3). It is implemented by a neural network which was designed by artificial evolution and tested with physical robots in previous works [5,4].
- `Assembly_Seed`. This behaviour is necessary to trigger the self-assembly process. If a red object is detected within 3 s of behaviour initiation, control is passed to `Aggregate` behaviour. (This prevents multiple seeding—if two nearby *s-bots* switch to `Assembly_Seed` behaviour, both will revert to `Aggregate` behaviour). After 3 s control is passed to `Group_Phototaxis` behaviour.

- `Group_Phototaxis`. The *s-bot* remains stationary if it detects blue objects in the vicinity (*s-bots* still assembling). Otherwise the *s-bot* performs phototaxis to the target. Because it is part of a *swarm-bot*, the orientation of the turret is fixed. The *s-bot* continually rotates the traction system with respect to the turret to keep the tracks oriented towards the target [5].

4 Results

We conducted a series of experiments in two different environments (see Fig. 3) with groups of 1, 2 and 3 *s-bots*.[1]

Trials with 3 *s-bots* in Environment A. We conducted 20 trials. In every trial all 3 *s-bots* reached the target zone. In 19 out the 20 trials the *s-bots* correctly navigated independently to the target. In a single trial the *s-bots* self-assembled on the down slope of the hill and then performed collective phototaxis to the target. The incorrect decision to self-assemble was due to a colour misperception of a non-existent object by an *s-bot*.

Trials with a single *s-bot* in Environment B. We modified the controller to only execute `Solo_Phototaxis` behaviour. The *s-bot* was thus limited to navigating towards the target taking no account of the terrain encountered.

We conducted 20 trials. The *s-bot* failed to overcome the hill in 20 out of 20 trials. In each trial the *s-bot* reached the hill and then toppled backwards due to the steepness of the slope.

To confirm that the *s-bot* was failing due to the intrinsic properties of the slope, we repeated this experiment at a number of different constant speeds.

Trials with 2 *s-bots* in Environment B. We conducted 20 trials. The *s-bots* successfully detected the slope in every trial. Furthermore the *s-bots* always succeeded in assembling into a 2 *s-bot swarm-bot*. In 13 trials (65%) the *swarm-bot* succeeded in overcoming the hill. In the other 7 trials (35%) the assembled *swarm-bot* failed to overcome the hill. These failures happened when the assembled *s-bots* attempted to climb the hill in parallel.

Trials with 3 *s-bots* in Environment B. We conducted 20 trials. The *s-bots* successfully detected the slope in every trial. In 16 trials (80%) all of the *s-bots* successfully self-assembled into a 3 *s-bot swarm-bot*. In each of these 16 trials the 3 *s-bot swarm-bot* went on to successfully reach the target area. Fig. 6 shows a sequence of images from a typical trial.

In the remaining 4 trials (20%) the *s-bots* still managed in each case to self-assemble into a *swarm-bot* of 2 *s-bots*. In two of these 4 trials the *swarm-bot* went on to successfully reach the target area. In the two other trials the *swarm-bot* was obstructed by the third *s-bot* which failed to self-assemble.

[1] Videos of all experiments can be found at `http://iridia.ulb.ac.be/~rogrady/ecal2005/`

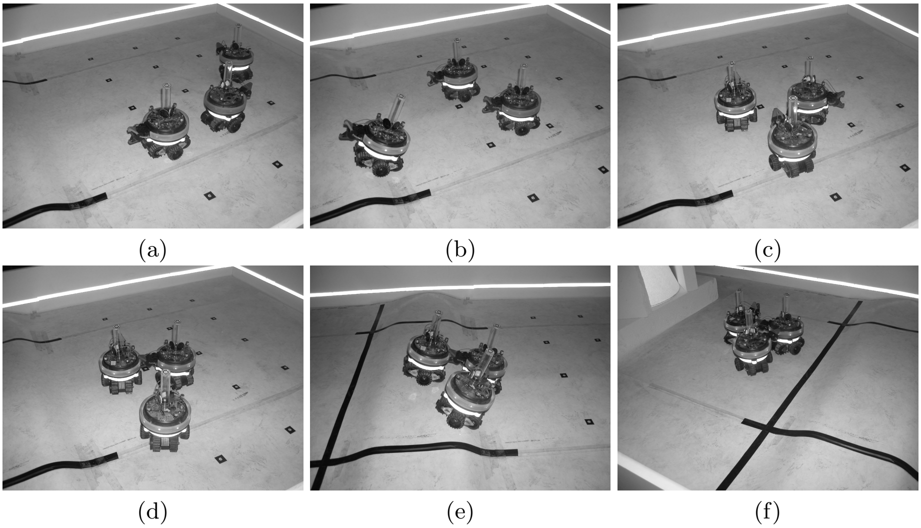


Fig. 6. The *s-bots* start in a random configuration (a). One *s-bot* detects a slope it cannot overcome alone and activates blue LEDs (b). The other *s-bots* detect blue colour (local communication). The group aggregates and self-assembles (c,d). The *s-bots* collectively overcome the rough terrain and reach the target area (e,f).

Table 1. Percentage of *s-bots* in Environment B trials succeeding for Self-assembly (**A**) and Completion of task (**C**)

	1 *s-bot* trials		2 *s-bot* trials		3 *s-bot* trials	
	A	**C**	**A**	**C**	**A**	**C**
% Successful (total)	-	0.00	100.00	65.00	93.33	86.67
% Successful alone	-	0.00	-	0.00	-	0.00
% Successful in 2 *s-bot swarm-bot*	-	-	100.00	65.00	13.33	6.67
% Successful in 3 *s-bot swarm-bot*	-	-	-	-	80.00	80.00
% Failed	-	100.00	0.00	35.00	6.67	13.33

4.1 Analysis

Table 1 shows the percentage of *s-bots* that successfully self-assembled and the percentage of *s-bots* that successfully completed the entire task in the Environment B experiments. The three columns distinguish between trials with 1 *s-bot*, 2 *s-bots* and 3 *s-bots*. The first row shows the total percentage of successful *s-bots*. Subsequent rows show the percentage of *s-bots* that were successful alone, or as part of a 2 *s-bot swarm-bot* or as as part of a 3 *s-bot swarm-bot*, or that failed.

The success rate for task completion increases with the number of robots. A single robot always fails. In 2 *s-bot* trials, 65% of *s-bots* complete the task. The 3 *s-bot* trials show a further clear improvement—86.67% complete the task.

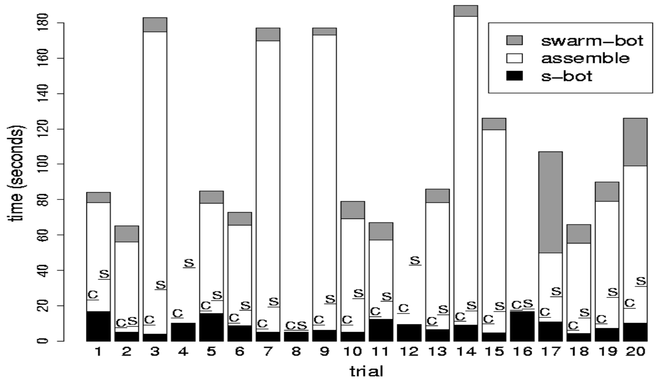


Fig. 7. 3 *s-bot* trials in Environment B. Phases represented are: (i) s-bot: independent *s-bot* navigation; (ii) assembly: aggregation and self-assembly; (iii) swarm-bot: collective *swarm-bot* navigation.

The fourth row (% Successful in 3 *s-bot swarm-bot*) shows that in the 3-*s-bot* trials 80% of *s-bots* successfully self-assemble into a 3 *s-bot swarm-bot*. The same row shows us that 80% of *s-bots* complete the task in a 3 *s-bot swarm-bot*. Thus in 3 *s-bot* trials, whenever all the 3 *s-bots* successfully self-assemble into a 3 *s-bot swarm-bot* they always successfully overcome the rough terrain. By contrast, in the 2 *s-bot* trials 100% of the *s-bots* self-assemble into a 2 *s-bot swarm-bot*. Despite this, only 65% of the 2 *s-bot swarm-bots* successfully overcome the hill.

The hill in environment B is such that in our trials a 3 *s-bot swarm-bot* always (100% of the trials) overcomes it. A 2 *s-bot swarm-bot* on the other hand sometimes (35% of the trials) fails to overcome the hill. Whenever the 2 *s-bot swarm-bot* approached the hill in parallel the *swarm-bot* toppled backwards.

Fig. 7 illustrates three phases of task completion. In the first phase (black segment) all *s-bots* are independently navigating to the target (this phase ends when the hill is first detected by an *s-bot*). The phase takes between 4 s and 17 s depending on the random initial configuration of the *s-bots*. For the unsuccessful trials (4,8,12,16) only this first phase is illustrated.

The second phase (white segment) consists of aggregation and self-assembly. This phase takes between 39 s and 175 s. This phase always accounts for a large percentage of total completion time due to its high level of complexity.

The final phase (grey segment) consists of collective phototaxis to the target. This phase takes between 4 s and 30 s, except in trial 17, when the *swarm-bot* got stuck for some time on the hill.

The symbol 'c' in Fig. 7 marks the first time that all *s-bots* become aware of the hill. In some trials (e.g. trials 5 and 6) the existence of the difficult hill is communicated very quickly between *s-bots* (see also Fig. 6). One *s-bot* detects the rough terrain and activates its blue ring LEDs. The other *s-bots* are already close enough to detect this blue colour. In such trials the point 'c' is reached soon after the start of the aggregation and self-assembly phase. In other trials (e.g. trials 1 and 12) it takes longer to reach point 'c' as the *s-bots* are sufficiently far apart that two *s-bots* discover the hill independently.

The symbol 's' in Fig. 7 indicates when self-assembly was seeded (the last time an *s-bot* switches to `Assembly_Seed` behaviour).

5 Conclusion

Self-assembly is a critical adaptive response mechanism in a number of social insect species. This work represents the first successful use of this response mechanism by real robots. We have shown that a group of physical autonomous mobile robots can choose to self-assemble in response to the demands of their task and environment. Using our controller, a group of robots faced with a simple hill overcome it independently. When the same robots are faced with a hill too difficult for a single robot they self-assemble and overcome the hill together. The success rate increased with the number of robots used: 0%,65% and 86.67% for groups of 1, 2 and 3 robots respectively.

Our approach involved splitting the task (as seen from the perspective of an individual robot) into distinct phases. Each phase was addressed by a separate behaviour module - these modules were combined to produce our behaviour based controller. In a previous work conducted in a simplified simulation environment, Trianni et al. [15] focused on evolving a single neural network controller to achieve functional self-assembly. We believe that application of this evolutionary approach to the real robots might yield solutions that exploit hidden properties of the robotic hardware or which make better use of the complex group dynamics of the task [13].

We are currently investigating mechanisms to generate connection patterns and group sizes [9] that are suited to particular tasks. In the spirit of functional self-assembly we would like the robots themselves to choose these patterns and group sizes as they interact with their environment.

Acknowledgements

This work was supported by the ANTS project, an *Action de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium; by the SWARM-BOTS project, funded by the Future and Emerging Technologies programme of the European Commission (grant IST-2000-31010); and by COMP2SYS, a Marie Curie Early Stage Research Training Site funded by the European Community's Sixth Framework Programme (grant MEST-CT-2004-505079). The information provided is the sole responsibility of the authors. It does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data in this publication. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Research Director.

References

1. C. Anderson, G. Theraulaz, and J.-L. Deneubourg. Self-assemblages in insects societies. *Insectes Soc.*, 49:99–110, 2002.
2. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, New York, NY, 1999.
3. M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L. M. Gambardella. Evolving self-organizing behaviors for a Swarm-bot. *Auton. Robots*, 17(2–3):223–245, 2004.
4. R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in mobile robotics. Technical Report IRIDIA/2005-2, IRIDIA - Université Libre de Bruxelles, 2005. Submitted to *IEEE Trans. Robot.*
5. R. Groß and M. Dorigo. Group transport of an object to a target that only some group members may sense. In *Parallel Problem Solving from Nature – 8th Int. Conf. (PPSN VIII)*, volume 3242 of *Lecture Notes in Computer Science*, pages 852–861. Springer Verlag, Berlin, Germany, 2004.
6. S. Hirose, T. Shirasu, and E. F. Fukushima. Proposal for cooperative robot "Gunryu" composed of autonomous segments. *Robot. Auton. Syst.*, 17:107–118, 1996.
7. A. Lioni, C. Sauwens, G. Theraulaz, and J.-L. Deneubourg. Chain formation in *Œcophylla longinoda. J. Insect Behav.*, 15:679–696, 2001.
8. A. Martinoli, K. Easton, and W. Agassounon. Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *Int. J. Robot. Res.*, 23(4):415–436, 2004.
9. C. Melhuish, O. Holland, and S. Hoddell. Convoying: Using chorusing to form travelling groups of minimal agents. *Robot. Auton. Syst.*, 28:207–216, 1999.
10. F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo. SWARM-BOTS: Physical interactions in collective robotics. *IEEE Robot. Autom. Mag.*, 2005, to appear.
11. F. Mondada, G. C. Pettinaro, A. Guignard, I. V. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. M. Gambardella, and M. Dorigo. SWARM-BOT: A new distributed robotic concept. *Auton. Robots*, 17(2–3):193–221, 2004.
12. S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji. M-TRAN: Self-reconfigurable modular robotic system. *IEEE/ASME Trans. Mechatron.*, 7(4):431–441, 2002.
13. S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines.* MIT Press, Cambridge, MA, 2000.
14. M. Rubenstein, K. Payne, P. Will, and W. M. Shen. Docking among independent and autonomous CONRO self-reconfigurable robots. In *Proc. of the 2004 IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2877–2882. IEEE Computer Society Press, Los Alamitos, CA, 2004.
15. V. Trianni, E. Tuci, and M. Dorigo. Evolving functional self-assembling in a swarm of autonomous robots. In *From Animals to Animats VIII. Proc. of the 8^{th} Inter. Conf. on Simulation of Adaptive Behavior*, pages 405–414. MIT Press, Cambridge, MA, 2004.
16. M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. B. Homans. Modular reconfigurable robots in space applications. *Auton. Robots*, 14(2-3):225–237, 2003.