On-Line Evolution of Controllers for Aggregating Swarm Robots in Changing Environments

Berend Weel[1], Mark Hoogendoorn[1], and A.E. Eiben[1]

Vrije Universiteit Amsterdam, The Netherlands
{b.weel,m.hoogendoorn,a.e.eiben}@vu.nl

Abstract. One of the grand challenges in self-configurable robotics is to enable robots to change their configuration, autonomously, and in parallel, depending on changes in the environment. In this paper we investigate, in simulation, if this is possible through evolutionary algorithms (EA). To this end, we implement an unconventional on-line, onboard EA that works inside the robots, adapting their controllers to a given environment on-line. This adaptive robot swarm is then exposed to changing circumstances that require that robots aggregate into "organisms" or dis-aggregate into swarm mode again to improve their fitness. The experimental results clearly demonstrate that this EA is capable of adapting the system in real time, without human intervention.

1 Introduction

Within the domain of self-configurable robotics, Stoy and Kurokawa [24] have identified a number of grand challenges, including one that a self-configurable robot *should be able to change its configuration, autonomously, and in parallel, depending on changes in the environment.* This is exactly the problem we address in this paper.

The main assumption and working hypothesis of the present study is that this problem can be solved by using evolutionary algorithms. Therefore, this paper falls in the area of evolutionary robotics, to be more specific in evolutionary swarm robotics, since we consider a swarm of robotic units that can physically aggregate and form a so-called organism, as envisioned by the Symbrion research project [16]. A specific feature of our system, that distinguishes it from the huge majority of related work, is that we use on-line evolution. In most evolutionary robotics systems the robot controllers are evolved off-line, *before* deploying the robots in some operational environment, cf. [20]. In contrast, we apply on-line evolution, *after* deployment, during the operational period of the robots. This feature is essential for robotic systems that are requested to operate long periods without direct human intervention, possibly in unforeseen and dynamically changing environments [19]. Our previous work has addressed the issue of self-driven aggregation and we have shown that even light environmental pressure is sufficient for the on-line evolution of aggregated organisms [27]. In this paper we switch from a static environment, as used in [27], to a dynamically changing one. The main research question is:

C.A. Coello Coello et al. (Eds.): PPSN 2012, Part II, LNCS 7492, pp. 245–254, 2012.

> Is our on-line evolutionary capable of repeatedly re-adapting the robot controllers if the circumstances change?

To find an answer to this question we design three different environments. One where aggregated organisms have an advantage, one where they have a disadvantage, and one that is neutral from this perspective. Then we expose a group of 50 robots to a scenario where the environment repeatedly changes and try to find out whether the organisms can adapt their sizes appropriately. To this end, there are two important things to note. Firstly, that the behaviour of organisms is the result of the behaviour of the individual robots that form their cells. Secondly, that robot controllers can only change through evolution and we do NOT use any specific fitness function to reward aggregation or disaggregation, only environmental selection.

2 Related Work

A seemingly related area of existing work is that of evolutionary optimisation in dynamic environments [4,18]. Our kind of on-line on-board evolutionary algorithms are similar to this because the actual (on-line) performance is more important than the end result (off-line performance). However, we are working with robots whose controllers need to be evolved on-the-fly (in vivo). Here lies a big difference: in our application one cannot afford bad candidate solutions, because they could "kill" the given robot, while in a usual EA bad individuals merely slow down the search.

Our work is related to both swarm robotics and self-reconfigurable modular robot systems. Swarm Robotics [17] is a field that stems from Swarm Intelligence [3], where swarm-robots often have the ability for physical self-assembly. Swarm-bots were created in order to provide a system which was robust towards hardware failures, versatile in performing different tasks, and navigating different environments. Similarly, self-reconfigurable modular robot systems were designed with three key motivations: versatility, robustness and low cost. The first two are identical to motivations for swarm-robots, while low cost can be achieved through economy of scales and mass production as these systems use many identical modules. Yim gives an overview of self-reconfigurable modular robot systems in [29], the research is mainly on creation of modules in hardware and showcasing their abilities to reconfigure and lift other modules. For our research, we assume a self-reconfigurable robot system which is independently mobile, as reported in [12,14,28]. The task of multiple robots connecting autonomously is usually called self-assembly, and has been demonstrated in several cases: [7,21,28,30]. Most of these however, are limited to pre-programmed control sequences without any evolution. In self-reconfigurable robots, self-assembly is restricted to the docking of two modules as demonstrated in [14,28].

On-Line On-Board evolution is a relatively new field in evolutionary robotics, initiated by the seminal paper of Watson *et al.* [26] who present a system where a population of physical robots (i.e. their controllers) autonomously evolves while situated in their task environment. Since then the area of on-line on-board evolution in swarm-robotics, as classified in [6], has gained a lot of momentum [5,10,12,13,22,25].

The work in this paper is part of the SYMBRION/REPLICATOR projects in which robots are being developed and used that are independently mobile and can operate as a swarm, but also have a mechanical docking mechanism allowing the modules to form and control a multi-robot organism [12]. The most closely related existing work is that of [2,7,8] that explores self-assembly of swarm robots. The controllers of the so-called s-bots (Recurrent Neural Networks) were evolved off-line in simulation, and deployed and tested in real s-bots afterwards. That research shows it is possible to evolve controllers which create organisms. Our present work is to demonstrate that it can be done through on-line evolution as a response to environmental changes.

3 System Description and Experimental Setup

As explained in the Introduction, we design three different environments. One where aggregated organisms have an advantage, one where they have a disadvantage, and one that is neutral from this perspective. Then we expose a group of 50 robots to a scenario where the environment repeatedly changes and watch whether they can adapt appropriately. In this section we describe the details.

Fig. 1. Overview of the final arena, consisting of neutral (white), organism-friendly (light-blue), and organism-unfriendly terrains

Arena. The main idea behind our implementation is to relate the environmental (dis)advantage of organisms to their ability to move and to use different terrains. To be specific, we add a "basic instinct" to the robots to move eastwards (from left to right in our arena) by defining their fitness through their positions: the more they move to the right during evaluation, the higher. Then we create three terrains that differ in their organism-friendliness. In the organism friendly terrain single robots cannot progress to the right and the speed of a larger organism is higher. Metaphorically speaking we have river with a west-bound current here, where only multi-cellular organism have the strength to swim eastwards. We make the organism unfriendly terrain by laying out narrow pathways where big organisms get stuck. The neutral terrain imposes no minimum nor maximum organism size. These three terrains are laid side by side and the resulting composed field is repeated three times in order to increase the number of environmental changes in one run, the resulting arena is shown in Fig. 1. Note that this arena is suited to test the populations response to changes, because robots are driven to move to the right by the fitness function. However, this fitness is certainly does not provide a specific reward for aggregating behaviour, thus it does not represent "cheating". In the meanwhile, it provides a well defined measure to assess success of robot behaviours: the more to the right at the end of an evaluation, the better.

Robots. We conduct our experiments with simulated e-puck robots in a simple 2D simulator: RoboRobo[1]. The robots can steer by setting their desired left and right wheel speeds. Each robot has 8 sensors to detect obstacles (static obstacles as well as other robots), as well as 8 sensors to detect the the river-like zones.

Connections. In our experiments robots can create new organisms, join an already existing organism, and two existing organisms can merge into a larger organism. When working with real robots, creating a physical connection between two robots can be challenging, and movements of joints are noisy because of actuator idiosyncrasies, flexibility of materials used, and sensor noise. We choose to disregard these issues and create a very simple connection mechanism which is rigid the moment a connection is made. The connection is modelled as a magnetic slip-ring, which a robot can set to 'positive', 'negative' or 'neutral'. When robots are close enough, they automatically create a rigid connection if both have their ring on the 'positive' setting. The connection remains in place as long a neither sets its slip-ring to 'negative'. Thus, a positive-neutral combination is not sufficient to establish a new connection, but it is sufficient to maintain an existing one. The neutral setting is important in this experiment to allow for organisms to maintain a certain size, as it allows connections to be maintained without creating new ones.

Controller. The controller is a feed-forward artificial neural network that selects one of 5 pre-programmed strategies based on sensory inputs. The neural net has 20 inputs (cf. Table 1), 8 outputs and no hidden nodes. It uses a *tanh* activation function. The inputs are normalised between 0 and 1.

The output of the neural network, as described in Table 1, is interpreted as follows: the first five outputs each vote for an action, the action with the highest activation level is selected. The sixth output describes the desired organism size which is used when the 'form organism' strategy is chosen. The seventh output describes the direction the robot should move in when performing the 'move' strategy. The eighth output is the desired speed the controller wants to move in, and is used in all strategies except 'halt' (which sets speed to 0).

Evolutionary Algorithm and Runs. We use an on-line on-board hybrid evolutionary mechanism. The first constituent of the hybrid is the $(\mu + 1)$ ON-LINE [9] method, where each robot is an island with a population of μ individuals (genotypes encoding possible controllers) that undergo evolution locally [1]. The other component is the peer-to-peer protocol based EVAG method [15]. The hybridised algorithm as described in detail in [11] also allows recombination across all robots in a panmictic overlay topology.

[1] `http://www.lri.fr/~bredeche/roborobo/`

Table 1. Neural Network inputs (left) and outputs (right)

8x Obstacle distance sensors	Vote for Form Organism
8x Zone distance sensors	Vote for Leave Organism
1x Size of the organism	Vote for Halt
1x Angle to the end	Vote for Avoid Obstacles
1x Distance to the end	Vote for Move
1x Bias node	Desired organism size
	Desired direction for Move
	Desired speed for Move

To represent robot controllers we use a genome which directly encodes the N weights of the neural net using a real-valued vector of length N. This genome is extended to include N mutation step sizes ($\sigma's$) for these N genes. Mutation is a standard Gaussian perturbation with noise drawn from $N(0, \sigma)$ using self-adaptation of $\sigma's$ through the standard formula's. For recombination we use averaging crossover. As for selection, we have a mixed system of global parent selection and local survivor selection. That is, parents are selected using a binary tournament over all genomes in all robots. Once the parents create a newborn controller its fitness is assessed by allowing it to control the robot for 1000 time steps: first a 'free' phase of 200 time steps to allow it to get out of bad situations, followed by an evaluation period for 800 time steps. Each 1000 time steps therefore constitutes 1 generation. At the start of a generation a choice is randomly made between creating a new controller as described above, or choosing an existing controller for re-evaluation, the chance of re-evaluating is controlled by the re-evaluation rate. At the end of the evaluation cycle the given controller is compared to the local population of μ others and replaces the worst one if it is better.

We ran the experiment using 50 robots, we used this number to have a relatively large amount of robots, while not over-crowding the starting area. Too many robots in the start area could lead to an inability of a controller to perform its otherwise good behaviour by getting stuck behind bad controllers.

We used the parameters shown in Table 2 for our evolutionary algorithm and repeated the experiment 50 times, each run lasting 2000 generations. The parameter settings are based on parameters found in our earlier paper [27] in which we used the BONESA toolbox[2] [23] to optimise settings for crossover rate, mutation rate, initial mutation step size, re-evaluation rate, and population size. Our experiments are fully repeatable, as the source code is available via the web-page of the first author[3].

Table 2. Parameters

Parameter	**Value**
Local population size	3
Mutation chance	0.4
Crossover chance	0.05
Re-evaluation rate	0.5
Initial mutation step-size	0.1
Generations	2000

[2] `http://sourceforge.net/projects/tuning/`

[3] A zip-file can be found at `http://www.few.vu.nl/~bwl400/papers/parcours.zip`

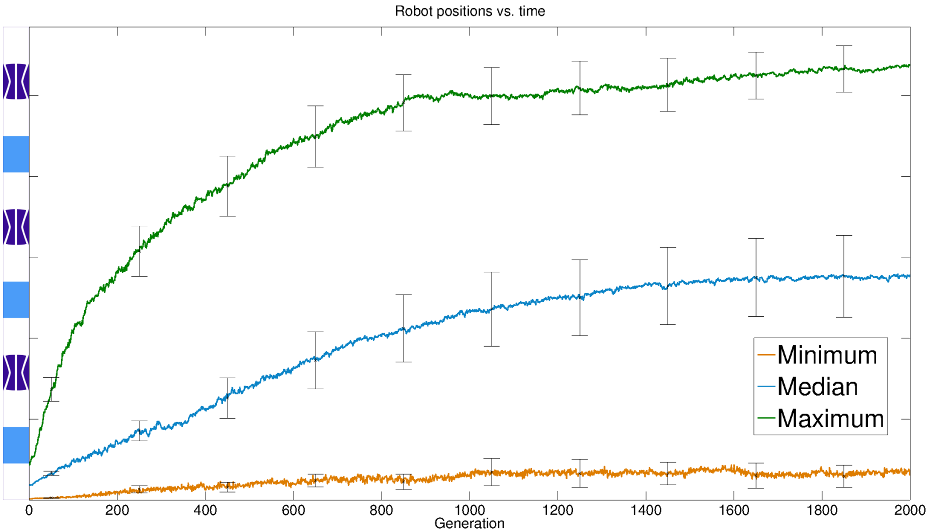

Fig. 2. Robot positions

4 Results and Analysis

This section presents the results of the experiment that have been performed. We essentially want to investigate (1) whether the robots are able to find their way through to the end the obstacle course, (2) analyse *how* they find their way through the obstacle course with respect to the formation of organisms. We will address both questions below.

4.1 Are They Able to Find Their Way?

In order to answer the first question, we have studied the positions of the robots within the obstacle course over time. Hereby, we have taken the position of the best performing robot during each run (i.e. the robot which came closest to the end of the obstacle course), and also recorded the position of the robot closest to the beginning of the course (the worst performing robot). Furthermore, we have taken the position of the median robot. The results averaged over 50 runs are shown in Figure 2. The layout of the obstacle course is shown on the y-axis whereas the x-axis show time (by means of the number of generations).

In the figure, it can be seen that the best robot on average is almost able to complete the entire obstacle course, meaning that it manages to pass the river three times, and ends up in the last narrow passageway. The reason why the best robots on average do not make it all the way to the end is due to the fact that there are some incidental bad runs where the best robot does not even pass the first obstacle.

When considering the worst individual, it can be seen that the worst performing robots hardly progresses within the obstacle course. On average the robots are not able to get beyond the first river which they encounter. In fact, they do

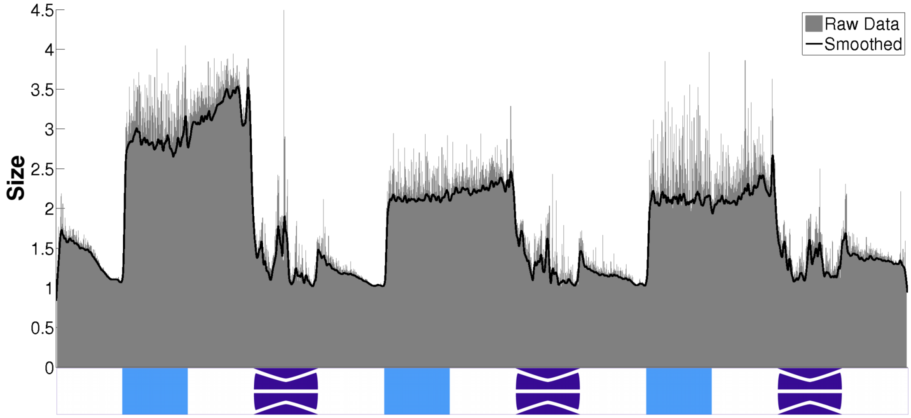

Fig. 3. Organism Size per Position

not even end up at the beginning of the river. The median robots manage to pass the first set of obstacles (the river and the narrow passageway) and are also able to pass the second river.

Overall, it can be concluded that on average a majority of the robots manage to find a way passed the river twice as well as a single passage of the narrow passageway. Some are able to do this three consecutive times. A minority of the robots is hindered too much by the obstacles, resulting in them never passing the first obstacle, namely the river.

4.2 How Do They Find Their Way?

It is interesting to see that the robots learn how to deal with the obstacles, but the question that remains is: how do they achieve it? Do they form organisms? And do they leave organisms? We will try to obtain some insights by studying the behaviour of the robots on a more detailed level. Therefore, we investigate the size of the organisms over the obstacle course to see whether they learn to form an organism and leave it at the appropriate locations.

Figure 3 shows the position in the obstacle course on the x-axis and the average organism size (a one denotes a single agent that is not part of an organism) on the y-axis.

In Fig. 3 can be seen that the average size of the organisms at the river area is a lot higher compared to the narrow passageway. In the neutral territories, the robots tend to continue with the organism size required by the obstacle they encountered last (i.e. after a river they remain within an

Table 3. Mean organism size in different zones. Zones are defined as the start and end of the river/narrow passage.

Zone	X region	Mean	Std
River 1	900–1800	2.76	0.45
Corridors 1	2700–3600	1.31	0.33
River 2	4500–5400	2.07	0.27
Corridors 2	6300–7200	1.28	0.27
River 3	8100–9000	2.07	0.35
Corridors 3	9900–10800	1.30	0.23

organism, and after the narrow passageway they remain single). When looking closer at the behaviour of the robots during the narrow passageway, a spike in the centre of the passageway can be seen. We assume that this is due to the fact that the curve in the passageway is difficult to pass for the robots, and therefore one option for them is to try and form an organism. In the trend of the organism size during the river passage it can be seen that the average size of the individuals is declining a bit after the first river. This because there are simply fewer robots around with which an organism can be formed, resulting in a disadvantage for robots that want to form large organisms.

Table 3 shows more detailed data on the average organism size at the various regions within the obstacle course. The standard deviations are also included.

5 Concluding Remarks and Further Research

In this paper we addressed the challenge of enabling a group of self-configurable robots to adapt their controllers to changing circumstances autonomously, without human intervention. The basic idea behind our approach is to equip the robots with evolutionary operators that keep working, during the operational period of the robots. Our algorithmic solution combines ideas from island-based EAs [1] and peer-to-peer EAs [15], offering –in principle– the best of both worlds.

Our experiments have provided convincing evidence that this approach is capable of evolving the robot controllers in real time and respond to environmental changes, without using a problem-tailored fitness function to "push" some targeted behaviour. Inevitably, we used a number of simplifying assumptions and design decisions in our experimental setup (e.g., using distance from the origin as an abstract measure of fitness), but these did not include any specific bias either. The emerging system behaviour was rooted in the interplay of the evolutionary mechanism and the environmental pressure.

Further work will be carried out in two directions. Firstly, we will explore the niche of applicability of our approach, by testing it in a number of different cases, i.e., in different (changing) environments, with various tasks for aggregated robots and measures of viability (fitness). One of the most interesting questions here concerns the combination of environmental selection (open-ended evolution for pure survival) and human-defined tasks (directed evolution with quantifiable performance measures). Secondly, in close cooperation with roboticists, we will port the whole machinery to real robots to validate its working *in vivo*.

Acknowledgements. This work was made possible by the European Union FET SYMBRION project. The authors would like to thank Evert Haasdijk and our partners in the SYMBRION consortium for many inspirational discussions.

References

1. Araujo, L., Merelo, J.: Diversity through multiculturality: Assessing migrant choice policies in an island model. IEEE Transactions on Evolutionary Computation 15(4), 456 – 469 (2011)

2. Bianco, R., Nolfi, S.: Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce. Connection Science 16(4), 227–248 (2004)
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press (1999)
4. Branke, J., Kirby, S.: Evolutionary Optimization in Dynamic Environments. Kluwer Academic Publishers, Boston (2001)
5. Bredeche, N., Montanier, J.-M.: Environment-Driven Embodied Evolution in a Population of Autonomous Agents. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 290–299. Springer, Heidelberg (2010)
6. Eiben, A.E., Haasdijk, E., Bredeche, N.: Embodied, on-line, on-board evolution for autonomous robotics. In: Levi, P., Kernbach, S. (eds.) Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution,, ch. 5.2, pp. 361–382. Springer (May 2010)
7. Groß, R., Bonani, M., Mondada, F., Dorigo, M.: Autonomous self-assembly in swarm-bots. IEEE Transactions on Robotics 22, 1115–1130 (2006)
8. Groß, R., Dorigo, M.: Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling. Adaptive Behavior 16(5), 285 (2008)
9. Haasdijk, E., Eiben, A.E., Karafotias, G.: On-line evolution of robot controllers by an encapsulated evolution strategy. In: Proceedings of the 2010 IEEE Congress on Evolutionary Computation. IEEE Computational Intelligence Society. IEEE Press, Barcelona (2010)
10. Hettiarachchi, S.: Distributed evolution for swarm robotics. ProQuest (2007)
11. Huijsman, R.J., Haasdijk, E., Eiben, A.: An On-line On-board Distributed Algorithm for Evolutionary Robotics. In: Proceedings of the 10th International Conference, Evolution Artificielle, EA 2011. LNCS. Springer (2011)
12. Kernbach, S., Meister, E., Scholz, O., Humza, R., Liedke, J., Rico, L., Jemai, J., Havlik, J., Liu, W.: Evolutionary robotics: The next-generation-platform for online and on-board artificial evolution. In: 2009 IEEE Congress on Evolutionary Computation, pp. 1079–1086 (2009)
13. König, L., Mostaghim, S., Schmeck, H.: Online and onboard evolution of robotic behavior using finite state machines. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, vol. 2, pp. 1325–1326. International Foundation for Autonomous Agents and Multiagent Systems (2009)
14. Kutzer, M.D.M., Moses, M.S., Brown, C.Y., Scheidt, D.H., Chirikjian, G.S., Armand, M.: Design of a new independently-mobile reconfigurable modular robot. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 2758–2764. IEEE (2010)
15. Laredo, J.L.J., Eiben, A.E., Steen, M., Merelo, J.J.: EvAg: a scalable peer-topeer evolutionary algorithm. Genetic Programming and Evolvable Machines 11(2), 227–246 (2009)
16. Levi, P., Kernbach, S. (eds.): Symbiotic Multi-Robot Organisms, Cognitive Systems Monographs, vol. 7. Springer, Heidelberg (2010)
17. Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I.W., Floreano, D., Deneubourg, J.L., Nolfi, S., Gambardella, L.M., Dorigo, M.: Swarm-bot: A new distributed robotic concept. Autonomous Robots 17(2/3), 193–221 (2004)
18. Morrison, R.W.: Designing Evolutionary Algorithms for Dynamic Environments. Springer (2004)
19. Nelson, A.L., Barlow, G.J., Doitsidis, L.: Fitness functions in evolutionary robotics: A survey and analysis. Robotics and Autonomous Systems 57(4), 345–370 (2009)

20. Nolfi, S., Floreano, D.: Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press, Cambridge (2000)
21. O'Grady, R., Christensen, A.L., Dorigo, M.: Autonomous Reconfiguration in a Self-assembling Multi-robot System. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217, pp. 259–266. Springer, Heidelberg (2008)
22. Schwarzer, C., Schlachter, F., Michiels, N.: Online evolution in dynamic environments using neural networks in autonomous robots. International Journal On Advances in Intelligent Systems 4(3-4), 288–298 (2012)
23. Smit, S.K., Eiben, A.E.: Multi-problem parameter tuning using BONESA. In: Hao, J., Legrand, P., Collet, P., Monmarché, N., Lutton, E., Schoenauer, M. (eds.) Proceedings of Artificial Evolution, 10th International Conference, Evolution Artificielle (EA 2011), pp. 222–233 (2011)
24. Stoy, K., Kurokawa, H.: Current topics in classic self-reconfigurable robot research. In: Proceedings of the 2011 IROS Workshop (SW9), Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems (2011)
25. Szymanski, M., Winkler, L., Laneri, D., Schlachter, F., Van Rossum, A., Schmickl, T., Thenius, R.: Symbricatorrtos: a flexible and dynamic framework for bio-inspired robot control systems and evolution. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 3314–3321. IEEE (2009)
26. Watson, R.A., Ficici, S.G., Pollack, J.B.: Embodied evolution: Distributing an evolutionary algorithm in a population of robots. Robotics and Autonomous Systems 39(1), 1–18 (2002)
27. Weel, B., Haasdijk, E., Eiben, A.E.: The Emergence of Multi-cellular Robot Organisms through On-Line On-Board Evolution. In: Di Chio, C., et al. (eds.) EvoApplications 2012. LNCS, vol. 7248, pp. 124–134. Springer, Heidelberg (2012)
28. Wei, H., Cai, Y., Li, H., Li, D., Wang, T.: Sambot: A self-assembly modular robot for swarm robot. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 66–71. IEEE (2010)
29. Yim, M., Shen, W.M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.S.: Modular self-reconfigurable robot systems [grand challenges of robotics]. IEEE Robotics & Automation Magazine 14(1), 43–52 (2007)
30. Yim, M., Shirmohammadi, B., Sastra, J., Park, M., Dugan, M., Taylor, C.: Towards robotic self-reassembly after explosion. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2767–2772. IEEE (2007)