



**Université Libre de Bruxelles**

*Institut de Recherches Interdisciplinaires  
et de Développements en Intelligence Artificielle*

## **Automatic off-line design of robot swarms: a manifesto**

M. BIRATTARI, A. LIGOT, D. BOZHINOSKI, M. BRAMBILLA, G. FRANCESCA,  
L. GARATTONI, D. GARZÓN RAMOS, K. HASSELMANN, M. KEGELEIRS,  
J. KUCKLING, F. PAGNOZZI, A. ROLI, M. SALMAN, and T. STÜTZLE

**IRIDIA – Technical Report Series**

Technical Report No.  
TR/IRIDIA/2019-002

April 2019  
Last revision: April 2019

**IRIDIA – Technical Report Series**  
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires  
et de Développements en Intelligence Artificielle*  
UNIVERSITÉ LIBRE DE BRUXELLES  
Av F. D. Roosevelt 50, CP 194/6  
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2019-002

Revision history:

TR/IRIDIA/2019-002.001	April 2019
TR/IRIDIA/2019-002.002	April 2019

The information provided is the sole responsibility of the authors and does not necessarily reflect the opinion of the members of IRIDIA. The authors take full responsibility for any copyright breaches that may result from publication of this paper in the IRIDIA – Technical Report Series. IRIDIA is not responsible for any use that might be made of data appearing in this publication.

---

# Automatic Off-line Design of Robot Swarms: A Manifesto

Mauro Birattari<sup>1,\*</sup>, Antoine Ligot<sup>1</sup>, Darko Bozhinoski<sup>1</sup>, Manuele Brambilla<sup>1</sup>,  
Gianpiero Francesca<sup>1</sup>, Lorenzo Garattoni<sup>1</sup>, David Garzón Ramos<sup>1</sup>, Ken  
Hasselmann<sup>1</sup>, Miquel Kegeleirs<sup>1</sup> Jonas Kuckling<sup>1</sup>, Federico Pagnozzi<sup>1</sup>,  
Andrea Roli<sup>2</sup>, Muhammad Salman<sup>1</sup>, and Thomas Stütze<sup>1</sup>

<sup>1</sup>*Université libre de Bruxelles, Belgium*

<sup>2</sup>*Alma Mater Studiorum, Università di Bologna, Italy*

Correspondence\*:

Mauro Birattari

mbiro@ulb.ac.be

## ABSTRACT

Designing collective behaviors for robot swarms is a difficult endeavor due to their fully distributed, highly redundant, and ever-changing nature. To overcome the challenge, a few approaches have been proposed, which can be classified as manual, semi-automatic, or automatic design. This paper is intended to be the manifesto of the automatic off-line design for robot swarms. We define the off-line design problem and illustrate it via a possible practical realization, highlight the core research questions, raise a number of issues regarding the existing literature that is relevant to the automatic off-line design, and provide guidelines that we deem necessary for a healthy development of the domain and for ensuring its relevance to potential real-world applications.

**Keywords:** swarm robotics, automatic design, collective behaviors, design methodology, evolutionary robotics

Although swarm robotics is widely recognized as a promising approach to coordinating large groups of robots (Dorigo et al., 2014; Yang et al., 2018), a general methodology for designing collective behaviors for robot swarms is still missing (Brambilla et al., 2013). The design problem is particularly challenging because it aims at producing a system that is autonomous, fully distributed, and highly redundant: robots do not have any predefined role and do not rely on any external infrastructure (Beni, 2004; Şahin, 2004). A robot swarm is a loosely coupled system in which the collective behavior of the system results from the local interactions between individuals, and between them and the environment. These interactions cannot be explicitly defined at design time due to the high uncertainty that characterizes the operation of a swarm. As a result, at least in the general case, it is impossible to tell what the individual robots should do so that a desired collective behavior is achieved. This rules out the application of traditional multi-robot systems and software engineering techniques, which rely on formally deriving the individual behaviors of the robots from specifications expressed at the collective level (Brugali, 2007; Di Ruscio et al., 2014; Bozhinoski et al., 2015; Schlegel et al., 2015).

A few methods/tools have been proposed that, under a number of restrictive hypotheses and constraints, support the designer for specific classes of missions (Hamann and Wörn, 2008; Kazadi, 2009; Berman et al., 2011; Beal et al., 2012; Brambilla et al., 2015; Reina et al., 2015; Lopes et al., 2016; Pinciroli and Beltrame, 2016). Also, a few automatic (and semi-automatic) design methods have been proposed that operate under various assumptions (Nolfi and Floreano, 2000; Francesca et al., 2014; Duarte et al., 2014). For a recent discussion, see Francesca and Birattari (2016).

This paper is intended to be the manifesto of the automatic off-line design of robot swarms. In this approach, the design problem is cast into an optimization problem that is solved off-line—that is, before the swarm is deployed in the target environment. An optimization algorithm searches a space of possible designs with the goal of maximizing an appropriate mission-specific performance measure. Within the design process, the performance of candidate designs explored by the optimization algorithm is assessed via computer-based simulations. Once the optimization algorithm terminates, the selected design is uploaded to the robots and the swarm is deployed in its target environment. In the following, we focus mostly on the development of software but the discussion can be directly extended to the hardware. For example, the automatic off-line design process might optimize the number of robots in the swarm; if the swarm is heterogeneous, select the fraction of robots of type A, B, C, . . . ; fine-tune parameters of hardware or firmware; activate/deactivate or add/remove hardware modules; design chassis, shell, or attachments.

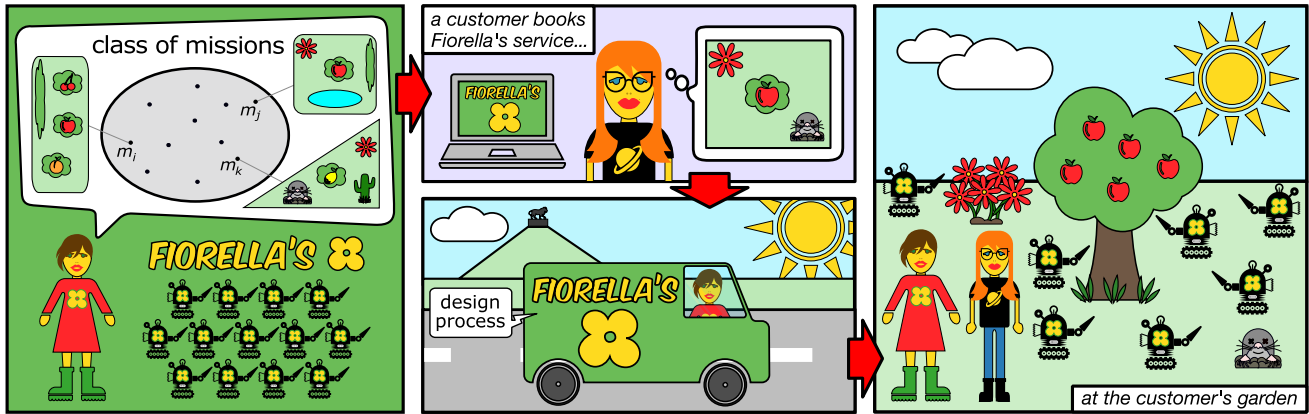
Our vision is that, in a relatively close future, automatic off-line design will be a practically relevant way of realizing robot swarms. Likely, it will not be the only one: other approaches will be available, each with its specific advantages and ideal areas of application, as well as its disadvantages and limitations. Among them, we foresee that a relevant role will be played by manual design, semi-automatic design, automatic on-line design, and hybrid approaches that combine the previous ones. Nonetheless, we expect that the automatic off-line approach will play a major role, both on its own and also as a component of hybrid approaches.

In the definition of automatic off-line design, a central role is played by the notion of class of missions: the set of missions of interest together with a probability measure defined on them, which determines their relative frequency of appearance. Indeed, we might be more likely asked to perform some missions and less likely others. Typically, an explicit, closed-form definition of the set of missions and of the probability measure is not available and is not even needed. What we have is a set or a stream of missions sampled from the class of interest according to the aforementioned probability measure. Assuming that missions are sampled according to a probability measure is what gives a formal meaning to the notion of expected performance, as well as to any other statistics one might wish to adopt to describe the aggregate behavior of a design method across the missions of interest. Indeed, an automatic off-line design method should be evaluated for its ability to realize robot swarms for missions sampled from a given class of interest, without any mission-specific adjustment, and without any per-mission human intervention. To clarify, it is convenient to sketch a possible practical realization of an automatic off-line design problem.

### **Fiorella's swarm gardening**

*(for an artist's rendition, see Fig. 1)*

*Fiorella owns a robot-swarm gardening business and offers her individually-tailored service to her many customers in the Brussels area. She has a busy schedule: every day, she visits three or four customers with her gardening swarm. Customers book Fiorella's service via a form on her website. Through the form, they ask for one or more specific interventions—e.g., cutting grass, watering flowers. They also provide relevant information on their garden—e.g., size, shape, orientation. The interventions requested and the characteristics of the garden specify the mission that Fiorella's swarm must perform for a specific customer. As the list of possible interventions and characteristics of the garden is huge, the class of possible missions is overwhelmingly large and rather diverse. To provide her customers with the best gardening experience—but also to cut costs and maximize her benefit—Fiorella relies on an automatic off-line method that designs and fine-tunes the behaviors of her swarm specifically*



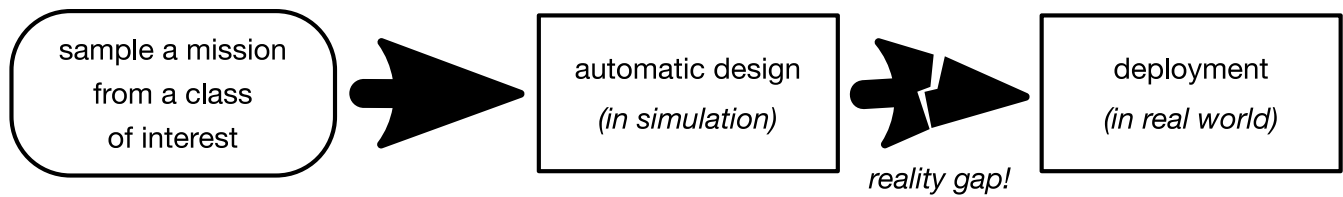
**Figure 1.** Fiorella’s swarm gardening. Fiorella’s robots can perform a large class of gardening missions. Through her website, customers book Fiorella’s services, specify the interventions to be performed, and provide a description of their garden. On the basis of this information, while Fiorella drives her robots to customers, her algorithms automatically design and fine-tune the behavior of the robots so as to offer a tailored service. When she arrives at a customer location, the gardening swarm is operational and immediately deployed.

*for each mission. The design process takes place while Fiorella drives her swarm to the customer’s garden: her powerful computers run simulations using the information provided by the customer on the interventions and on the garden. The design process must be performed within a limited amount of time—the time of the ride to the customer’s. As Fiorella arrives on the spot, the selected design is uploaded to the robots and the swarm is deployed in the garden. Fiorella cannot intervene in the design process—she drives the van in the meantime. Moreover, due to her tight schedule, Fiorella cannot either test the selected design on the robots before deployment and possibly re-run the design process: once she reaches the customer, robots must be operational. Any per-mission human intervention and any test on the robots in the target environment would be too time consuming and expensive: they would increase costs dramatically and Fiorella would be unavoidably out of business.*

Missions in the class of interest can be relatively minor variations of each other—e.g., cut the grass in a small garden; in a large one; in one with a central flower bed. In this case, the behaviors to be produced will be similar, with some minor differences to increase performance or reduce execution time. Missions can also be substantially different in the nature of their goals and require major differences in the behaviors to be realized—e.g., cut the grass; gather dead leaves in a specific place; locate and map mole tunnels. In Fiorella’s example, the central role of the notion of class of missions emerges clearly. Fiorella faces a stream of missions sampled from the possible missions for which customers might demand her intervention—and which her swarm can hopefully accomplish. It is in the repetitive nature of the design problems faced by Fiorella that the significance of automatic design lies. Indeed, if Fiorella had to solve a single design problem (instead of a stream thereof) she could profitably solve it either manually or via a semi-automatic design method.<sup>1</sup> It is only when one has to solve a stream of design problems that the human intervention might become uneconomical or even unfeasible. Conceiving, implementing, and setting up an automatic design method is in itself an investment of time and resources, which pays off only if the design process is

<sup>1</sup> By semi-automatic design, we mean an approach in which a human designer is assisted by an optimization process similar to the one of automatic design, but can afford intervening in the process (on a per-mission basis) to guide it according to their insight.

Repeat *ad libitum*:



**Figure 2.** Flowchart diagram of the automatic off-line design process. A mission is sampled from a class of interest. Using computer-based simulation, an automatic design method defines a robot swarm tailored to the sampled mission. Once the automatic design terminates, the swarm is deployed in the target environment. The process is repeated *ad libitum*. One of the most challenging issues in automatic off-line design is the reality gap—the subtle but inevitable difference between simulation and reality (Brooks, 1992; Jakobi et al., 1995).

then repeated a sufficient number of times on multiple missions—those of the class for which the design process is conceived. If one had to address a single mission, it would be more sensible to invest time and resources on the specific mission rather than on a more general automatic design method that addresses a whole class of missions. For a schematic representation of the automatic off-line design process, see Fig. 2.

Fiorella's example allows us to highlight a number of issues and research questions that are relevant to the automatic off-line design of robot swarms. Can effective robot swarms be designed automatically via an off-line process? Can we conceive a method to automatically design a swarm for any mission within a given class? What is the class of missions for which a given method can design an appropriate swarm? How can a given method be generalized to solve a larger class of mission? Given a class of missions, which is the most appropriate design method? What elements/characteristics of a design method influence its ability to handle missions of a given class? Vice versa, what elements/characteristics of a class of missions might suggest that a given method is appropriate to handle them? Which features of a specific mission make it challenging for a given design method? Are these challenging features equally challenging for all possible design methods? Is it possible to match challenging features of missions with characteristics of design methods? To what extent a given design method is robust to the so called reality gap—that is, the difference between simulation models and reality? Is it possible to predict the performance drop that a swarm designed off-line will experience when deployed in the target environment? Are different design methods equally sensitive to the reality gap? What elements/characteristics of a design method make it more or less robust to the reality gap? Can these characteristics be leveraged to engineer a design method that is inherently robust to the reality gap? What is the best way to devise models to be used within an off-line design process? How efficient is a design method? In other terms, how many off-line simulation runs are required to produce an effective design? What elements/characteristics of a design method increase/decrease its efficiency? How well does a given design method behave for a large/small design budget—that is, when allowed to perform few/many off-line simulation runs? Does the efficiency of a design method depend on the specific mission or class of missions considered? What elements/characteristics of a mission determine the minimum size of the design budget needed to produce an effective design? Which design method is more effective when the design budget is small/large? When should a design process be stopped? This list of questions encompasses many relevant issues but it is by no means exhaustive. For example, other relevant issues would concern the off-line definition of the swarm size (or its spatial density), its impact on performance, and the robustness/scalability of behaviors that are automatically designed.

Although a body of literature exists that is relevant to the automatic off-line design of robot swarms, it is our contention that the core research questions have not been properly addressed, yet. The largest share of the design methods described in the relevant literature belong in the neuro-evolutionary domain (Nolfi and Floreano, 2000): robots are controlled by a neural network whose synaptic weights (and possibly the topology) are optimized by an evolutionary algorithm (Quinn et al., 2003; Christensen and Dorigo, 2006; Baldassarre et al., 2007; Trianni, 2008; Hauert et al., 2009; Trianni and Nolfi, 2009; Waibel et al., 2009; Ferrante et al., 2013; Gomes et al., 2013; Ferrante et al., 2015; Trianni and López-Ibáñez, 2015). For a review of the neuro-evolutionary approach see Floreano and Keller (2010), Bongard (2013), Bongard and Lipson (2014), Doncieux and Mouret (2014), Trianni (2014), and Silva et al. (2016). Other approaches depart from neuro-evolution as (i) robots are controlled by software architectures other than neural networks (Hecker et al., 2012; Gauci et al., 2014a), or (ii) they adopt optimization algorithms other than an evolution algorithm (Pugh et al., 2005), or (iii) both (Francesca et al., 2014; Gauci et al., 2014b; Francesca et al., 2015; Kuckling et al., 2018).

Unfortunately, with only few exceptions, the aforementioned methods have been studied following protocols that were not conceived to address the core research questions sketched above. In almost the totality of the studies, the focus is on a specific mission that must be performed by a swarm—or, equivalently, on a specific capability that the swarm should acquire and display. The design method is proposed only as a way to achieve the desired collective behavior and is not the protagonist of the study: the study is not structured to highlight its properties and assess its performance. The design method has so little importance that it is not customarily given an identifying name—contrary to what happens in related fields such as machine learning or heuristic optimization. Typically, the design method is tested on a single mission and it is not compared to any alternative. It is extremely rare that a same design method is tested across multiple studies on multiple missions without undergoing any (manually-applied) mission-specific modification. Rarely, the control software produced by a design method is tested on robots to assess whether and to what extent it crosses the reality gap satisfactorily. Moreover, the design method at hand will survive only the time span of the paper in which it has been introduced and its implementation will not be made publicly available for further studies, to be possibly performed by a third party. Typically, a design method is run iteratively on a single mission. It is run once, the behavior generated is inspected by the designer who then modifies the method itself or the objective function to be optimized—e.g., by adding/removing terms. These activities are then iterated at will until a satisfactory behavior is obtained. In most cases, this iterative process is not detailed in research articles: it is often unclear how many iterations have been performed, what has been measured at each iteration, what modifications have been implemented, what ideas have been tried and then abandoned. Research articles typically present only the final setting that eventually generated the behavior discussed. The iterative process is repeated only once, as it would be difficult to produce independent trials of a process that features a human in the loop. As a result, research articles fail to inform on the robustness and repeatability of the process. With this, we do not mean to say that the protocols we criticize above are wrong, meaningless, or useless. Indeed, they were defined to address some interesting questions that, although not fully relevant to the automatic off-line design of robot swarms, are definitely relevant to other domains including, for example, evolutionary biology and the semi-automatic design of robot swarms.

An appropriate protocol to address the aforementioned issues should reflect the following tenets of the research in automatic off-line design: (i) automatic off-line design methods should not be mission-specific and should be able to address a whole class of missions without undergoing any modification; (ii) once a mission is specified, human intervention is not provided for in any phase of the design process. Indeed, research that is intended to be relevant to the automatic off-line design of robot swarms should exclude

the case in which design methods are conceived for or are manually adapted to a specific mission—for example, by manually tuning parameters of the optimization algorithm and/or of the control architecture, or by pre-filtering sensor readings on the basis of insight that only a human designer can provide. It should also exclude the case in which, on a per-mission basis, human designers are allowed to inspect (via either simulation or robot experiments) the behavior of an automatically designed swarm and, on the basis of their observations, modify elements of the automatic design process and iterate it at will, until they obtain satisfactory results. In particular, human designers should not be allowed, on a per-mission basis, to use any insight gained through inspection to modify the design method (optimization algorithm, architecture, sensor pre-filtering, etc.); to adapt simulation models; and to amend the objective function by adding/removing terms so as to steer the design process as wished. On the other hand, to effectively contribute to the development of the domain, researchers in the automatic off-line design of robot swarms should pay particular attention to a number of methodological issues. In particular, they should: (a) provide a clear and thorough description of the design methods they propose, including a list of the value of all parameters; (b) precisely characterize the platforms for which these methods can generate control software; (c) clearly identify and name methods for future reference; (d) publish implementations; (e) test methods on multiple missions; (f) identify—at least informally—the class of missions that a method is intended to address; (g) perform comparative studies in which methods under analysis are tested under the same conditions; and (h) run robot experiments to assess robustness to the reality gap. It is our contention that this minimal set of guidelines will allow the domain to grow healthy and thriving so as to eventually prove its practical relevance in real-world applications.

### Author Contributions

All authors contributed to the elaboration of the ideas presented in the paper, read the text, and provided comments. In particular, MB started the discussion and lead it; he also drafted the paper and coordinated its revision. AL selected/reviewed the relevant literature and contributed to the formulation of the research questions. DB contributed to framing the complexity of the design problem in swarm robotics. The other authors equally contributed and are listed in alphabetical order.

### Funding

The project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 681872). MB, JK, and TS acknowledge support from the Belgian *Fonds de la Recherche Scientifique* – FNRS. DGR acknowledges support from the Colombian *Administrative Department of Science, Technology and Innovation* – COLCIENCIAS. AR acknowledges the support of the *Chaire internationale* programme of Université libre de Bruxelles.

### REFERENCES

- Baldassarre, G., Trianni, V., Bonani, M., Mondada, F., Dorigo, M., and Nolfi, S. (2007). Self-organized coordinated motion in groups of physically connected robots 37, 224–39. doi:10.1109/TSMCB.2006.881299
- Beal, J., Dulman, S., Usbeck, K., Viroli, M., and Correll, N. (2012). Organizing the aggregate: Languages for spatial computing. In *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments* (IGI Global), chap. 16. 436–501. doi:10.4018/978-1-4666-2092-6.ch016



- Beni, G. (2004). From swarm intelligence to swarm robotics. In *Swarm Robotics, SAB* (Berlin, Germany: Springer), vol. 3342 of *LNCS*, 1–9. doi:10.1007/978-3-540-30552-1\_1
- Berman, S., Kumar, V., and Nagpal, R. (2011). Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination. In *IEEE International Conference on Robotics and Automation – ICRA*, ed. L. Zexiang (Piscataway, NJ: IEEE), 378–385
- Bongard, J. and Lipson, H. (2014). Evolved machines shed light on robustness and resilience. *Proceedings of the IEEE* 102, 899–914. doi:10.1109/JPROC.2014.2312844
- Bongard, J. C. (2013). Evolutionary robotics. *Commun. ACM* 56, 74–83. doi:10.1145/2493883
- Bozhinoski, D., Di Ruscio, D., Malavolta, I., Pelliccione, P., and Tivoli, M. (2015). Flyaq: Enabling non-expert users to specify and generate missions of autonomous multicopters (Piscataway, NJ: IEEE), 801–806. doi:10.1109/ASE.2015.104
- Brambilla, M., Brutschy, A., Dorigo, M., and Birattari, M. (2015). Property-driven design for swarm robotics: A design method based on prescriptive modeling and model checking 9, 17.1–28. doi:10.1145/2700318
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective 7, 1–41. doi:10.1007/s11721-012-0075-2
- Brooks, R. (1992). Artificial life and real robots. In *Towards a Practice of Autonomous Systems. Proceedings of the First European Conference on Artificial Life* (Cambridge, MA: MIT Press), 3–10
- Brugali, D. (2007). *Software engineering for experimental robotics*, vol. 30 (Springer). doi:10.1007/978-3-540-68951-5
- Christensen, A. L. and Dorigo, M. (2006). Evolving an integrated phototaxis and hole-avoidance behavior for a swarm-bot. In *Artificial Life – ALIFE* (Cambridge, MA: MIT Press), 248–254
- Di Ruscio, D., Malavolta, I., and Pelliccione, P. (2014). A family of domain-specific languages for specifying civilian missions of multi-robot systems. In *Proceedings of the 1st International Workshop on Model-Driven Robot Software Engineering – MORSE*. 13–26
- Doncieux, S. and Mouret, J.-B. (2014). Beyond black-box optimization: a review of selective pressures for evolutionary robotics 7, 71–93. doi:10.1007/s12065-014-0110-x
- Dorigo, M., Birattari, M., and Brambilla, M. (2014). Swarm robotics. *Scholarpedia* 9, 1463. doi:10.4249/scholarpedia.1463
- Duarte, M., Oliveira, S., and Christensen, A. (2014). Evolution of hierarchical controllers for multirobot systems. In *Artificial Life – ALIFE* (Cambridge, MA: MIT Press), 657–664. doi:10.7551/978-0-262-32621-6-ch105
- Ferrante, E., Duéñez Guzmán, E., Turgut, A. E., and Wenseleers, T. (2013). Geswarm: Grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics. In *Genetic and Evolutionary Computation – GECCO* (New York, NY: ACM). 17–24. doi:10.1145/2463372.2463385
- Ferrante, E., Turgut, A., Duéñez-Guzmán, E., Dorigo, M., and Wenseleers, T. (2015). Evolution of self-organized task specialization in robot swarms 11, e1004273. doi:10.1371/journal.pcbi.1004273
- Floreano, D. and Keller, L. (2010). Evolution of adaptive behaviour in robots by means of darwinian selection 8, 1–8. doi:10.1371/journal.pbio.1000292
- Francesca, G. and Birattari, M. (2016). Automatic design of robot swarms: achievements and challenges 3, 1–9
- Francesca, G., Brambilla, M., Brutschy, A., Garattoni, L., Miletitch, R., Podevijn, G., et al. (2015). AutoMoDe-Chocolate: automatic design of control software for robot swarms 9, 125–152. doi:10.1007/s11721-015-0107-9

- Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., and Birattari, M. (2014). AutoMoDe: A novel approach to the automatic design of control software for robot swarms 8, 89–112. doi:10.1007/s11721-014-0092-4
- Gauci, M., Chen, J., Li, W., Dodd, T. J., and Groß, R. (2014a). Clustering objects with robots that do not compute. In *Autonomous Agents and Multiagent Systems – AAMAS* (Richland, SC: IFAAMAS), 421–428
- Gauci, M., Chen, J., Li, W., J Dodd, T., and Groß, R. (2014b). Self-organized aggregation without computation 33, 1145–1161. doi:10.1177/0278364914525244
- Gomes, J., Urbano, P., and Christensen, A. (2013). Evolution of swarm robotics systems with novelty search 7, 115–144. doi:10.1007/s11721-013-0081-z
- Hamann, H. and Wörn, H. (2008). A framework of space–time continuous models for algorithm design in swarm robotics 2, 209–239. doi:10.1007/s11721-008-0015-3
- Hauert, S., Zufferey, J.-C., and Floreano, D. (2009). Evolved swarming without positioning information: an application in aerial communication relay 26, 21–32. doi:10.1007/s10514-008-9104-9
- Hecker, J. P., Letendre, K., Stolleis, K., Washington, D., and Moses, M. E. (2012). Formica ex machina: ant swarm foraging from physical to virtual and back again. In *Swarm Intelligence – ANTS* (Berlin: Springer), vol. 7461 of *LNCS*, 252–259
- Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: the use of simulation in evolutionary robotics. In *Advances in Artificial Life* (London, UK: Springer), vol. 929 of *LNCS*, 704–720
- Kazadi, S. (2009). Model independence in swarm robotics 2, 672–694. doi:10.1108/17563780911005836
- Kuckling, J., Ligot, A., Bozhinoski, D., and Birattari, M. (2018). Behavior trees as a control architecture in the automatic modular design of robot swarms. In *Swarm Intelligence – ANTS* (Cham, Switzerland: Springer), vol. 11172 of *LNCS*. 30–43
- Lopes, Y. K., Trenkwalder, S. M., Leal, A. B., Dodd, T. J., and Groß, R. (2016). Supervisory control theory applied to swarm robotics 10, 65–97. doi:10.1007/s11721-016-0119-0
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics* (Cambridge, MA: MIT Press)
- Pinciroli, C. and Beltrame, G. (2016). Buzz: A programming language for robot swarms. *IEEE Software* 33, 97–100. doi:10.1109/MS.2016.95
- Pugh, J., Martinoli, A., and Zhang, Y. (2005). Particle swarm optimization for unsupervised robotic learning. In *Swarm Intelligence Symposium – SI* (Piscataway, NJ: IEEE), 92 – 99. doi:10.1109/SIS.2005.1501607
- Quinn, M., Smith, L., Mayley, G., and Husbands, P. (2003). Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors 361, 2321–43. doi:10.1098/rsta.2003.1258
- Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., and Trianni, V. (2015). A design pattern for decentralised decision making. *PLoS ONE* 10, e0140950. doi:10.1371/journal.pone.0140950
- Şahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. In *Swarm Robotics, SAB* (Berlin, Germany: Springer), vol. 3342 of *LNCS*, 10–20
- Schlegel, C., Lotz, A., Lutz, M., Stampfer, D., Inglés-Romero, J. F., and Vicente-Chicote, C. (2015). Model-driven software systems engineering in robotics: covering the complete life-cycle of a robot. *It-Information Technology* 57, 85–98. doi:10.1515/itit-2014-1069
- Silva, F., Duarte, M., Correia, L., Oliveira, S., and Christensen, A. (2016). Open issues in evolutionary robotics. *Evolutionary Computation* 24, 205–236. doi:10.1162/EVCO\_a\_00172
- Trianni, V. (2008). *Evolutionary Swarm Robotics* (Berlin, Germany: Springer)
- Trianni, V. (2014). Evolutionary robotics: model or design? 1, 1–6. doi:10.3389/frobt.2014.00013

- Trianni, V. and López-Ibáñez, M. (2015). Advantages of task-specific multi-objective optimisation in evolutionary robotics. *PLoS ONE* 10, e0136406. doi:10.1371/journal.pone.0136406
- Trianni, V. and Nolfi, S. (2009). Self-organizing sync in a robotic swarm: a dynamical system view 13, 722–741. doi:10.1109/TEVC.2009.2015577
- Waibel, M., Keller, L., and Floreano, D. (2009). Genetic team composition and level of selection in the evolution of multi-agent systems 13, 648–660. doi:10.1109/TEVC.2008.2011741
- Yang, G.-Z., Bellingham, J., E. Dupont, P., Fischer, P., Floridi, L., Full, R., et al. (2018). The grand challenges of Science Robotics. *Science Robotics* 3, eaar7650. doi:10.1126/scirobotics.aar7650