Disentangling automatic and semi-automatic approaches to the optimisation-based design of control software for robot swarms

Mauro Birattari, Antoine Ligot, and Ken Hasselmann
IRIDIA, Université libre de Bruxelles, Belgium

Abstract

Optimisation-based design is an effective and promising approach to realising collective behaviours for robot swarms. Unfortunately, the domain literature remains often vague on the exact role played by the human designer, if any. It is our contention that two cases should be disentangled: semi-automatic design, in which a human designer operates and steers an optimisation process (e.g., by fine-tuning the parameters of the optimisation algorithm); and (fully) automatic design, in which the optimisation process does not involve, need, or allow any human intervention. In the paper, we briefly review the relevant literature, we illustrate the hypotheses, the characteristics, and the core challenges of semi-automatic and automatic design, and we sketch the context in which they could be ideally applied.

Swarm robotics [1] is a promising approach to controlling large groups of autonomous robots [2]. Although it has attained a notable position in the scientific literature [3, 4, 5, 6, 7, 8, 9], the lack of a general methodology for designing collective behaviors for robot swarms hinders its real-world application [10]. In a swarm, robots do not have predefined roles, nor do they rely on external infrastructures; they act based on local information collected through their sensors or relayed by their neighbouring peers. The result is a loosely coupled system whose behaviour emerges from the interactions between the individuals, and between the individuals and the environment. It is indeed the decentralised nature of a robot swarm that makes the design problem particularly challenging. The collective behaviour of a swarm cannot be directly programmed: it can only be obtained by specifying the appropriate individual behaviour, which is generally a difficult endeavour due to the uncertainty that characterises the operation of a swarm. Which robot interacts with which other or with which feature of the environment, and when this happens, is the result of how the system evolves and is unknown at design time.

To obtain a desired collective behaviour, designers usually proceed by trial and error. Unfortunately, traditional multi-robot systems and software engineering techniques cannot be applied to swarm robotics. They rely on the formal derivation of the individual behaviour from specifications expressed at the collective level [11, 12, 13, 14]. Such a derivation cannot be performed in the general case because of the decentralised nature of a robot swarm and of the related issues highlighted above. A few principled manual design methods have been proposed [15, 16, 17, 18, 19, 20, 21, 22, 23], but their working hypotheses and constraints limit their general applicability.

In the rest of the paper, we discuss the use of optimisation-based methods for designing robot swarms: we distinguish between semi-automatic and automatic design, we highlight their potential applications, and discuss issues that we deem crucial for their future development.

Optimisation-based design of robot swarms

An important share of the research in swarm robotics has been dedicated to what we could call *optimisation-based design*. In optimisation-based design, the mission to be accomplished by the swarm is formally specified by defining a performance measure, a function that evaluates the extent to which the swarm attains the goals and/or violates the constraints of the mission. The whole design problem is formulated as an optimisation problem: the possible individual behaviours are the search space explored by an optimisation algorithm that maximises an appropriate objective function. The objective function could be the aforementioned performance measure or any other function that, besides measuring the degree of success in performing the mission, includes the (prior) knowledge of a human expert on how the mission could be tackled by the swarm. A familiar approach to optimisation-based design is neuro-evolutionary swarm robotics, in which each robot is controlled by a neural network that maps sensor readings to actuator commands [24]. The parameters and possibly the structure of the neural network are selected via an evolutionary algorithm that maximises an objective function, which in the evolutionary parlance is known as *fitness function*.

The distinction we make here between manual and optimisation-based design is not to be intended as a rigid taxonomy but rather as a convenient way of understanding and reasoning on the nature of different approaches. The two categories should not be intended as mutually exclusive. Indeed, hybrids are possible and even promising. For example, the control software produced manually could be refined by an optimisation algorithm to fine-tune free parameters better than a human could do. Although the optimisation algorithm might significantly contribute to improving the performance, its role would remain secondary and the overall structure of the control software eventually produced would remain the one defined by the human designer. This would have the benefit of preserving intrinsic structural properties of the solution, which could be used to guarantee properties such as stability and convergence, whenever relevant [17]. Another possible context in which it is desirable to preserve the structure of the solution is when one wishes to identify the parameters of models that reproduce behaviours observed in biological systems [25].

In this paper, we restrict our attention to the design of robot swarms for two main reasons: swarm robotics is our domain of expertise and, as mentioned above, the design problem in swarm robotics raises specific issues that make design by optimisation particularly interesting. Nonetheless, we believe that the reasoning that follows can be relevant to other areas of robotics and can be reformulated, at least in its main traits, also for the optimisation-based design of single-robot systems.

Two categorisations of optimisation-based methods

Optimisation-based design methods can be analysed and understood on the basis of different categorisations. A commonly adopted categorisation builds on the notions of on-line and off-line design [26, 27]. In on-line design, the design process is distributed and operates on the robots while they perform their mission. On-line design is promising for specific applications—for example, the real-time adaptation of a few parameters of the control software to track non-stationary features of the environment. Nonetheless, it does not appear to be the ultimate solution to designing robot swarms. Its drawbacks include that (i) it can handle a relatively small search space; (ii) it can be applied only to cases in which the robots are able to evaluate their collective performance; and (iii) sub-optimal instances of control software—typically explored early in the optimisation process—can cause damage to the robots and/or to the environment. In off-line design, the design process is performed before the swarm is deployed. Typically, off-line design relies on computer simulations that reproduce the relevant features of the robots and of the target environment. In addition to simulations, tests can be performed with the target robotics platform (or with a simplified version) in a mock-up environment that reproduces the relevant features of the one in which the robots will eventually operate. As tests with physical robots are typically much more costly and time-consuming than simulations, off-line design relies mostly on the latter. Moreover, simulations are not affected by the aforementioned drawbacks of on-line design. Indeed, (i) being faster and cheaper than robot experiments, simulation allows exploring a larger space of solutions; (ii) simulation provides a *God-eye view* on the swarm, which allows optimising also objective functions that could not be computed by individual robots; (iii) simulation prevents that robots and the environment are damaged in the design process. However, by (fully or partially) relying on simulation, off-line design faces a problem that does not affect on-line design: the so-called *reality gap* [28, 29], which is the possibly subtle, yet unavoidable, difference between simulation and reality. Due to the reality gap, control software generated in simulation suffers a performance drop when deployed on physical robots [30, 31]. Also in the case of the on-line/off-line categorisation, hybrids are possible: one could use off-line design to generate a system in which some parameters are selected for being fine-tuned on-line, after deployment.

A further possible categorisation of optimisation-based design stands on the notions of semi-automatic design and automatic (or, more explicitly, fully automatic) design. This second categorisation, which is orthogonal to the on-line/off-line one, is less commonly encountered in the domain literature and it is very rare that authors explicitly position their contribution with respect to it. Yet, we find this categorisation particularly enlightening and of fundamental importance for properly framing the research questions to be investigated and for defining appropriate experimental protocols to address them. The on-line/off-line and the automatic/semi-automatic categorisations are orthogonal in the sense that they are based on different, unrelated criteria. By crossing them, four categories can be highlighted: semi-automatic off-line design, semi-automatic on-line design, automatic off-line design, and automatic on-line design. Although the four categories are possible, they have not received the same amount of attention so far. In the literature, the most represented category is the one of semi-automatic off-line design, followed by semi-automatic on-line and automatic off-line design. To the best of our knowledge, no automatic on-line design method has been proposed and convincingly assessed, yet. References to the relevant literature are given below, when semi-automatic and automatic design are illustrated. As stated above, the on-line/off-line categorisation has been already discussed in the literature and appears to be well understood by the community. In particular, the implications that the notions of on-line and off-line design have on the definition of the respective research questions and experimental protocols are clear and, in our opinion, do not require further discussion at the moment. On the contrary, it is our contention that the automatic/semi-automatic categorisation has not received sufficient attention. For this reason, in the following of our discussion, we will exclusively focus on it.

Semi-automatic and automatic design

In semi-automatic design, a human designer operates an optimisation algorithm as its main design tool. In the off-line case, semi-automatic design is typically an iterative process in which the designer, guided by their in-

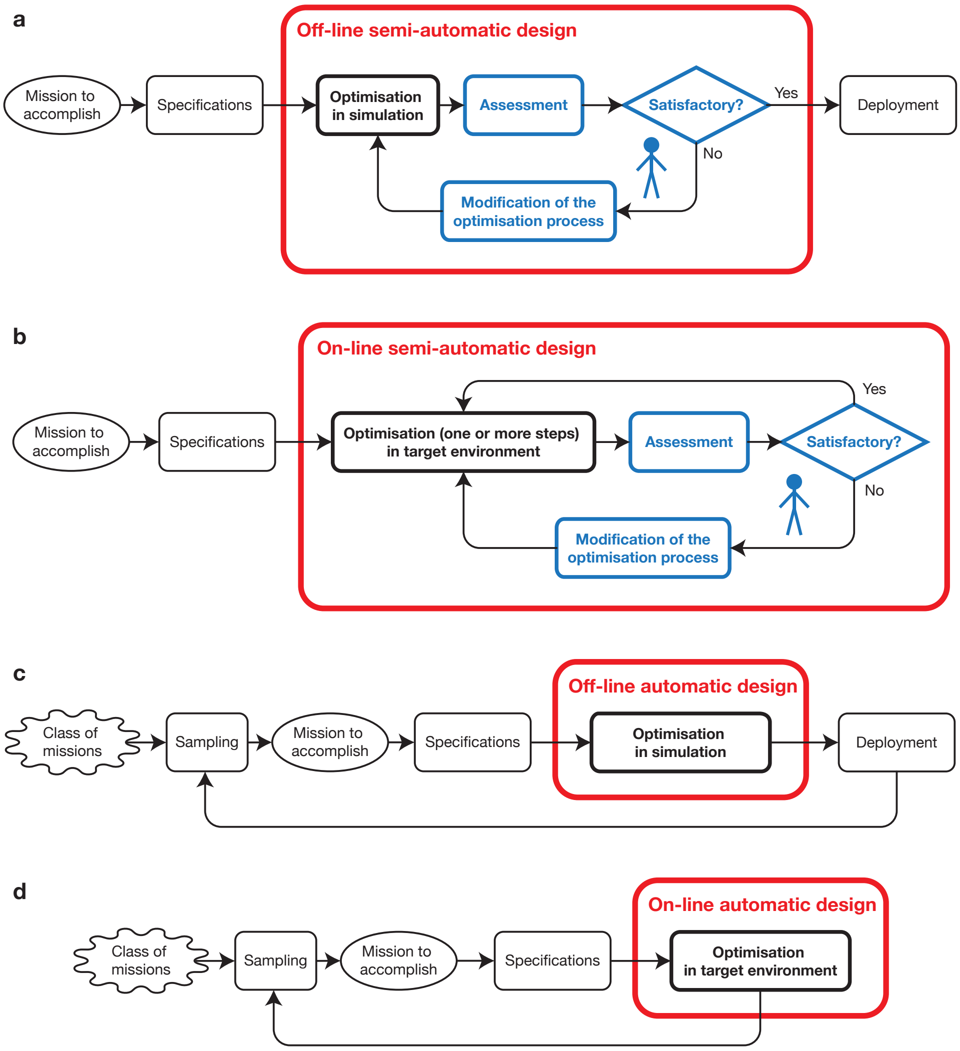

Figure 1: **Flowcharts of typical approaches to optimisation-based design. a.** Off-line semi-automatic design. **b.** On-line semi-automatic design. **c.** Off-line automatic design. **d.** On-line automatic design. In the flowcharts, the design process as such is the part contained in the red box. The blocks outside the red box represent other phases of the life cycle: mission selection/sampling, specifications, and deployment. They are represented here to provide the context in which optimisation-based design takes place. A semi-automatic design process—be it off-line or on-line—is always an iterative process characterised by a feedback loop: it relies on an optimisation algorithm, but features a human designer in the loop.

tuition and previous experience, instantiates a first optimisation process on the basis of their understanding of the mission the swarm must accomplish. Then, they evaluate the behaviour produced by the optimisation process either in simulation or with physical robots. On the basis of this evaluation, the designer modifies the optimisation process in a way that, according to their judgement, is expected to yield control software that would allow the swarm to perform the mission more effectively. The elements of the optimisation process that might be modified include characteristics of the control software architecture (e.g., if a neural network is adopted, number of hidden neurons/layers, presence of recurrent connections), the prefiltering of sensor readings, the encoding of input and output, the simulation models, the parameters of the optimisation algorithm (e.g., if an evolutionary algorithm is adopted, population size, mutation/crossover probability), and the objective function that is optimised (e.g., adding or removing terms to penalise/reward the emergence of certain behavioural features). The designer executes the updated version of the optimisation process for then evaluating the new behaviour produced. These steps are repeated until the designer is satisfied with the control software obtained and/or feels that it cannot be improved any further. All in all, the process relies on an optimisation algorithm, but features a human designer in the loop (Fig. 1a). In the on-line case, semi-automatic design is typically performed by conceiving the optimisation process and manually tailoring it to the mission to be accomplished. Also in the on-line case, the process has often an iterative nature: the designer observes an issue, makes a diagnosis and an hypothesis on what could be modified to prevent the issue, and modifies the process accordingly (Fig. 1b). Both in the off-line and on-line case, the precise moments in which the human designer intervenes, the exact nature of their intervention, and the level of expertise that the human designer must have might vary and are specific to each semi-automatic design method. Semi-automatic design methods could be therefore characterised and in principle (partially) ranked according to the quantity/quality of human intervention they require. Most of the studies presented in the neuro-evolutionary swarm robotics literature [24, 26] belong in the semi-automatic approach. In fact, little details is typically provided on role of the experimenter and no explicit mention is typically made of how many iterations of the optimisation process were needed to converge to the final setting that is eventually discussed in the article. This is a long-known issue and, already back in 2006, Christensen & Dorigo noted that: "It is not customary for authors in the field of evolutionary robotics to disclose how the chosen evolutionary setup was found. We believe that many evolutionary setups are found in an ad-hoc fashion" [32]. Indeed, in the neuro-evolutionary swarm robotics literature, the design process is typically manually tailored to the specific mission at hand and, as such, qualifies as semi-automatic design. It relies on the fact that human designers incorporate prior mission-specific knowledge in the design process [33] and often "struggle with the choice of suitable (fitness) functions using a trial-and-error strategy" [34]—see also the notions of *incremental evolution* and *fitness shaping* [35] and the one of *human-in-the-loop* [36]. Moreover, in the neuro-evolutionary swarm robotics literature, empirical evaluations are typically focused on designing control software for a single mission and are not therefore conceived to tell whether and to what extent the method discussed can be applied out-of-the-box (and therefore in a fully automatic way, without any further manual adaptation) to any other mission besides the one considered. It remains that, mainly thanks to the results achieved using the neuro-evolutionary approach, the literature shows that semi-automatic design is an effective way to realise robot swarms—both off-line [37, 38, 39, 32, 40, 24, 41, 42, 43] and on-line [44, 45, 46]. The downside is that semi-automatic design is a labour-intensive process and demands the attention of an expert designer: the decisions that need to be taken require a good understanding of the design process and of the mission at hand. As a direct result of the essential involvement of a human expert in the design process and of the judgemental nature of the decisions they make, results are often hardly reproducible.

In automatic design, the optimisation process is performed in a fully automatic way and does not provide for any per-mission intervention of a human designer: once the mission is specified—notably, by defining an appropriate performance measure—any human intervention is proscribed [47]. An automatic design method is expected to be able to produce control software for a whole class of missions—characterised by different goals, constraints and therefore different performance measures—without the need to undergo any per-mission manual modification (Fig. 1c-d). To the best of our knowledge, only few works have been devoted to the automatic design of robot swarms and have tested design methods on multiple missions without applying any per-mission manual modification [48, 49, 50, 51, 52, 53]. All these methods belong in off-line design.

A few other works exist that describe optimisation-based methods that could possibly qualify as automatic, either off-line [54, 55, 56, 57] or on-line [58]. The authors of these works do not describe any human intervention within the design process, which would support the conclusion that these methods are indeed automatic. Yet, the authors do not explicitly state either that the optimisation process has not been defined/fine-tuned iteratively on the basis of results observed on the specific mission at hand: if it has, the methods would rather qualify as semi-automatic. In any case, these works focus on a single mission and therefore do not provide evidence that the methods presented can be applied to other missions without the need to undergo any per-mission, manual modification.

Semi-automatic design: consider an international project involving hundreds of engineers aiming at realising a swarm-based construction system that will build an outpost on Mars. The mission that the robots will eventually have to perform is complex. Sufficient time and resources are available for repeating the design process multiple times, adjusting parameters, testing the resulting design in simulation and also in an Earth-based mock-up environment reproducing the conditions that the robots will face once deployed on Mars.

Automatic design: consider a small one-person business providing gardening services using a robot swarm. The control software of the swarm is automatically designed and fine-tuned specifically for each intervention. Each single intervention is relatively simple, but customers might require a rather large and varied spectrum of services to be performed. Tight time and monetary constraints rule out the possibility of designing and fine-tuning the control software by hand or via semi-automatic design, testing the control software produced, and performing multiple iterations of the design process.

Figure 2: **Potential applications of semi-automatic and automatic design.**

Potential applications of semi-automatic and automatic design

In our vision, both semi-automatic and automatic design are relevant to the development of swarm robotics; they will occupy two different niches and will address different contexts of application. Examples of potential applications of semi-automatic and automatic design are given in Fig. 2.

Semi-automatic design is appealing to handle an individual, complex design problem for a specific mission that would be too difficult or time-consuming to be solved manually by a human designer via trial and error. Given the exceptional, one-of-a-kind nature of such a design problem, it is reasonable to assume that (i) a human designer can supervise a semi-automatic design process to produce custom-made control software for the mission at hand and (ii) it is not worth developing a fully automatic design method that would be then used only once to generate control software for the single mission of interest.

Automatic design is particularly appealing when a design process is to be executed repeatedly on different missions belonging to a given class and it is impossible, impractical, or economically unfeasible that a human designer performs, supervises, or checks the design process itself. Under these conditions, developing a method that is able to produce control software in a fully automatic way is the ideal solution. Such a method should be able to produce control software for any mission within the class of interest and must do so without any human intervention nor any per-mission modification or adjustment. Indeed, any manual adaptation of an automatic design method would defeat the very purpose of automatic design.

In both semi-automatic and automatic design, the control software produced must be robust to variations of the operating conditions as these are not perfectly known at design time. Per-mission robustness can be achieved by running the design process on randomised variants of the original mission. Although different, these variants are formally specified by the same performance measure and do not qualify as a class of missions.

The core challenge faced by semi-automatic and automatic design is different. In semi-automatic design, the challenge lies in the complexity of the single mission for which control software must be generated. Conversely, in automatic design, the challenge rather lies in the complexity of the class itself—that is, in the variety and diversity of the missions comprised therein.

As semi-automatic and automatic design adopt similar or even the same tools and optimisation algorithms, there is a clear overlap between the two research domains. They share a number of research questions, and advances in one of them has likely implications for the other. Examples of shared research questions are those concerning optimisation algorithms, software architectures used to encode the behaviour of the individual robots, or the simulation models adopted in the design process. Yet, each of the two domains raises specific research questions that characterise it. For example, questions concerning the role of the human designer are specific to semi-automatic design, while those regarding the class of missions that can be solved by a method are specific to automatic design. To address the specific research questions, empirical studies in the two domains must adopt different protocols. For example, a protocol is inappropriate in automatic design if it focuses on the generation of control software for a single mission and/or allows a human designer to run repeatedly a design process, assess the control software produced, and manually modify and fine-tune the optimisation algorithm, the control architecture, and/or the sensor prefiltering. Conversely, such a protocol would be perfectly appropriate in semi-automatic design. To be appropriate to study automatic design methods, a protocol needs to assess their ability to generate control software without any per-mission human intervention. The methods must therefore be tested on multiple missions without undergoing any manually applied per-mission adjustment.

Although the literature on the optimisation-based design of robot swarms clearly indicates that both automatic and semi-automatic design are promising, the results obtained so far have the nature of feasibility studies and many important issues are still to be addressed. We are convinced that the next essential step that the research community must take is to establish a state of the art. So far, little attention has been devoted to the systematic empirical analysis and comparison of the various ideas and methods that have been proposed. Both in automatic and semi-automatic design, the research community needs to focus on defining appropriate experimental protocols. This is particularly challenging in semi-automatic design, where the presence of a human in the loop makes objective comparisons problematic. In this case, a specific protocol needs to be adopted that involves possibly large samples of human subjects so as to avoid any bias that would be introduced by relying on a single experimenter, who could have preferences for one method or for another. A crucial element in an experimental protocol is the definition of benchmarks that mimic the specific context in which semi-automatic and automatic design are expected to be applied. A further element is the definition of evaluation criteria. An obvious criterion to evaluate a design method is the performance of the robot swarm produced, but others might be relevant: for example, the amount of resources needed by a method, and its usability. These criteria might need to be specified differently for semi-automatic and automatic design. In semi-automatic design, the relevant resources to be measured include the typical number of iterations needed, the computation time, and the amount of time that the human designer must devote to the design process. The usability could be measured by the level of experience that the human designer needs to have to operate it or by the time needed to train a human expert to its use. In automatic design, the most relevant resource is the computation time, and the usability of a method could be measured by the effort needed to implement it or port it to a different robotics platform. Besides considering the aforementioned criteria in isolation, research in the optimisation-based design of robot swarms should also explore the various trade-offs existing between them. Indeed, for example, a method could yield high performance, but require significant resources; or could be easy to use, but inappropriate if top-tier performance is required.

Disentangling semi-automatic and automatic design is crucial to properly frame the future research in the design by optimisation of robot swarms. We are convinced that the clear understanding of the specificities of semi-automatic and automatic design will allow the community to properly state the relevant research questions and to define appropriate experimental protocols to address them. It will also contribute to set correct and realistic expectations on what each of the two approaches could and should produce.

References

[1] Dorigo, M., Birattari, M. & Brambilla, M. Swarm robotics. *Scholarpedia* **9**, 1463 (2014).

[2] Yang, G.-Z. *et al.*. The grand challenges of Science Robotics. *Sci. Robot.* **3**, eaar7650 (2018).

[3] Rubenstein, M., Cornejo, A. & Nagpal, R. Programmable self-assembly in a thousand-robot swarm. *Science* **345**, 795–799 (2014).

[4] Werfel, J., Petersen, K. & Nagpal, R. Designing collective behavior in a termite-inspired robot construction team. *Science* **343**, 754–758 (2014).

[5] Garattoni, L. & Birattari, M. Autonomous task sequencing in a robot swarm. *Sci. Robot.* **3**, eaat0430 (2018).

[6] Slavkov, I. *et al.*. Morphogenesis in robot swarms. *Sci. Robot.* **3**, eaau9178 (2018).

[7] Yu, J., Wang, B., Du, X., Wang, Q. & Zhang, L. Ultra-extensible ribbon-like magnetic microswarm. *Nat. Commun.* **9**, 3260 (2018).

[8] Li, S. *et al.*. Particle robotics based on statistical mechanics of loosely coupled components. *Nature* **567**, 361–365 (2019).

[9] Xie, H. *et al.*. Reconfigurable magnetic microrobot swarm: Multimode transformation, locomotion, and manipulation. *Sci. Robot.* **4**, eaav8006 (2019).

[10] Brambilla, M., Ferrante, E., Birattari, M. & Dorigo, M. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* **7**, 1–41 (2013).

[11] Brugali, D. (Ed.) *Software Engineering for Experimental Robotics* (Springer, Berlin, Germany, 2007).

[12] Di Ruscio, D., Malavolta, I. & Pelliccione, P. 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* (eds Aßmann, U. & Wagner, G.) **1319** (CEUR-WS, Aachen, Germany, 2014), 13–26.

[13] Bozhinoski, D., Di Ruscio, D., Malavolta, I., Pelliccione, P. & Tivoli, M. Flyaq: enabling non-expert users to specify and generate missions of autonomous multicopters. In *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (eds Cohen, M., Grunske, L. & Whalen, M.) (IEEE, Piscataway, NJ, USA, 2015), 801–806.

[14] Schlegel, C. *et al.*. Model-driven software systems engineering in robotics: covering the complete life-cycle of a robot. *it – Inf. Technol.* **57**, 85–98 (2015).

[15] Hamann, H. & Wörn, H. A framework of space–time continuous models for algorithm design in swarm robotics. *Swarm Intell.* **2**, 209–239 (2008).

[16] Kazadi, S. Model independence in swarm robotics. *Int. J. Intell. Comput. Cybern.* **2**, 672–694 (2009).

[17] Berman, S., Kumar, V. & Nagpal, R. Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination. In *IEEE International Conference on Robotics and Automation, ICRA* (IEEE, Piscataway, NJ, USA, 2011), 378–385.

[18] Beal, J., Dulman, S., Usbeck, K., Viroli, M. & Correll, N. Organizing the aggregate: languages for spatial computing. In *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments* (ed Marjan, M.) 436–501 (IGI Global, Hershey, Pennsylvania, USA, 2012).

[19] Brambilla, M., Brutschy, A., Dorigo, M. & Birattari, M. Property-driven design for swarm robotics: a design method based on prescriptive modeling and model checking. *ACM Tran. Auton. Adap.* **9**, 17:1–17:28 (2014).

[20] Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M. & Trianni, V. A design pattern for decentralised decision making. *PLoS One* **10**, e0140950 (2015).

[21] Lopes, Y. K., Trenkwalder, S. M., Leal, A. B., Dodd, T. J. & Groß, R. Supervisory control theory applied to swarm robotics. *Swarm Intell.* **10**, 65–97 (2016).

[22] Pinciroli, C. & Beltrame, G. Buzz: A programming language for robot swarms. *IEEE Softw.* **33**, 97–100 (2016).

[23] Hamann, H. *Swarm Robotics: A Formal Approach* (Springer, Cham, Switzerland, 2018).

[24] Trianni, V. *Evolutionary Swarm Robotics* (Springer, Berlin, Germany, 2008).

[25] Hecker, J. P., Letendre, K., Stolleis, K., Washington, D. & Moses, M. E. Formica ex machina: ant swarm foraging from physical to virtual and back again. In *International Conference on Swarm Intelligence – ANTS 2012* (eds Dorigo, M. *et al.*) **7461** (Springer, Berlin, Germany, 2012), 252–259.

[26] Francesca, G. & Birattari, M. Automatic design of robot swarms: achievements and challenges. *Front. Robot. AI* **3**, 1–9 (2016).

[27] Bredeche, N., Haasdijk, E. & Prieto, A. Embodied evolution in collective robotics: a review. *Front. Robot. AI* **5**, 12 (2018).

[28] Brooks, R. A. Artificial life and real robots. In *Towards a Practice of Autonomous Systems. Proceedings of the First European Conference on Artificial Life* (eds Varela, F. J. & Bourgine, P.) (MIT Press, Cambridge, MA, USA, 1992), 3–10.

[29] Jakobi, N., Husbands, P. & Harvey, I. Noise and the reality gap: the use of simulation in evolutionary robotics. In *Advances in Artificial Life: Third european conference on artificial life* (eds Moraán, F., Moreno, A., Merelo, J. J. & Chacón, P.) **929** (Springer, Berlin, Germany, 1995), 704–720.

[30] Nolfi, S. & Floreano, D. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines.* (MIT Press, Cambridge, MA, USA, 2000).

[31] Floreano, D., Husbands, P. & Nolfi, S. Evolutionary robotics. In *Springer Handbook of Robotics* (eds Siciliano, B. & Khatib, O.), 1423–1451 (Springer, Berlin, Heidelberg, Germany, 2008).

[32] Christensen, A. L. & Dorigo, M. Evolving an integrated phototaxis and hole-avoidance behavior for a swarm-bot. In *Artificial Life – ALIFE* (eds Rocha, L. M. *et al.*) (MIT Press, Cambridge, MA, USA, 2006), 248–254.

[33] Divband Soorati, M. & Hamann, H. The effect of fitness function design on performance in evolutionary robotics: The influence of a priori knowledge. In *GECCO 2015: Proceedings of the Genetic and Evolutionary Computation Conference* (ed Silva, S.) (ACM, New York, NY, USA, 2015), 153–160.

[34] Floreano, D. & Urzelai, J. Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Netw.* **13**, 431–443 (2000).

[35] Doncieux, S. & Mouret, J.-B. Beyond black-box optimization: a review of selective pressures for evolutionary robotics. *Evol. Intell.* **7**, 71–93 (2014).

[36] Silva, F., Duarte, M., Correia, L., Oliveira, S. M. & Christensen, A. L. Open issues in evolutionary robotics. *Evol. Comput.* **24**, 205–236 (2016).

[37] Quinn, M., Smith, L., Mayley, G. & Husbands, P. Evolving controllers for a homogeneous system of physical robots: structured cooperation with minimal sensors. *Philos. Trans. R. Soc. A* **361**, 2321–2343 (2003).

[38] Dorigo, M. *et al.*. Evolving self-organizing behaviors for a Swarm-bot. *Auton. Robots* **17**, 223–245 (2003).

[39] Ampatzis, C., Tuci, E., Trianni, V. & Dorigo, M. Evolving communicating agents that integrate information over time: a real robot experiment. In *Artificial Evolution. Seventh International Conference, Evolution Artificielle (EA 2005)* (eds Talbi, E. G., Liardet, P., Collet, P., Lutton, E. & Schoenauer, M.) **3871** (Springer, Berlin, Germany, 2006), 248–254.

[40] Trianni, V. & Dorigo, M. Self-organisation and communication in groups of simulated and physical robots. *Biol. Cybern.* **95**, 213–231 (2006).

[41] Ampatzis, C., Tuci, E., Trianni, V., Christensen, A. L. & Dorigo, M. Evolving self-assembly in autonomous homogeneous robots: experiments with two physical robots. *Artif. Life* **15**, 465–484 (2009).

[42] Duarte, M. *et al.*. Evolution of collective behaviors for a real swarm of aquatic surface robots. *PLoS One* **11**, e0151834 (2016).

[43] Jones, S., Studley, M., Hauert, S. & Winfield, A. Evolving behaviour trees for swarm robotics. In *Distributed Autonomous Robotic Systems* (eds Groß, R. *et al.*) **6** (Springer, Cham, Switzerland, 2016), 487–501.

[44] Watson, R. A., Ficici, S. G. & Pollack, J. B. Embodied evolution: distributing an evolutionary algorithm in a population of robots. *Robot. Auton. Syst.* **39**, 1–18 (2002).

[45] Bredeche, N., Montanier, J.-M., Liu, W. & Winfield, A. Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. *Math. Comput. Model. Dyn. Syst.* **18**, 101–129 (2012).

[46] Jones, S., Winfield, A., Hauert, S. & Studley, M. Onboard evolution of understandable swarm behaviors. *Adv. Intell. Syst.* **1**, 1900031 (2019).

[47] Birattari, M. *et al.*. Automatic off-line design of robot swarms: a manifesto. *Front. Robot. AI* **6**, 59 (2019).

[48] Waibel, M., Keller, L. & Floreano, D. Genetic team composition and level of selection in the evolution of multi-agent systems. *IEEE Trans. Evol. Comput.* **13**, 648–660 (2009).

[49] Francesca, G., Brambilla, M., Brutschy, A., Trianni, V. & Birattari, M. AutoMoDe: a novel approach to the automatic design of control software for robot swarms. *Swarm Intell.* **8**, 89–112 (2014).

[50] Francesca, G. *et al.*. An experiment in automatic design of robot swarms: AutoMoDe-Vanilla, EvoStick, and human experts. In *Swarm Intelligence – ANTS* (eds Dorigo, M. *et al.*) **8667** (Springer, Berlin, Germany, 2014), 25–37.

[51] Francesca, G. *et al.*. AutoMoDe-Chocolate: automatic design of control software for robot swarms. *Swarm Intell.* **9**, 125–152 (2015).

[52] Hasselmann, K., Robert, F. & Birattari, M. Automatic design of communication-based behaviors for robot swarms. In *Swarm Intelligence – ANTS* (eds Dorigo, M. *et al.*) **11172** (Springer, Cham, Switzerland, 2018), 16–29.

[53] Kuckling, J., Ligot, A., Bozhinoski, D. & Birattari, M. Behavior trees as a control architecture in the automatic modular design of robot swarms. In *Swarm Intelligence – ANTS* (eds Dorigo, M. *et al.*) **11172** (Springer, Cham, Switzerland, 2018), 30–43.

[54] Baldassarre, G. *et al.*. Self-organized coordinated motion in groups of physically connected robots. *IEEE Trans. Syst. Man Cybern. B* **37**, 224–239 (2007).

[55] Trianni, V. & Nolfi, S. Self-organizing sync in a robotic swarm: a dynamical system view. *IEEE Trans. Evol. Comput.* **13**, 722–741 (2009).

[56] Gauci, M., Chen, J., Li, W., Dodd, T. J. & Groß, R. Clustering Objects with Robots That Do Not Compute. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems – AAMAS2014* (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, USA, 2014), 421–428.

[57] Gauci, M., Chen, J., Li, W., Dodd, T. J. & Groß, R. Self-organized aggregation without computation. *Int. J. Robot. Res.* **33**, 1145–1161 (2014).

[58] Usui, Y. & Arita, T. Situated and embodied evolution in collective evolutionary robotics. In *Proceedings of the 8th International Symposium on Artificial Life and Robotics* (2003), 212–215.

Acknowledgements

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). M.B. acknowledges support from the Belgian *Fonds de la Recherche Scientifique*–FNRS, of which he is a Research Director.

Author contributions

M.B. lead the discussion, drafted the manuscript, and coordinated its revision. All authors contributed to the elaboration of the ideas presented in the manuscript, read it, and provided comments.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence should be addressed to M.B.