Automatic Design of Communication-Based Behaviors for Robot Swarms

Ken Hasselmann[1], Frédéric Robert[2], and Mauro Birattari[1](✉)

[1] IRIDIA, Université Libre de Bruxelles, Brussels, Belgium
{khasselm,mbiro}@ulb.ac.be
[2] BEAMS, Université Libre de Bruxelles, Brussels, Belgium
frrobert@ulb.ac.be

Abstract. We introduce Gianduja, an automatic design method that generates communication-based behaviors for robot swarms. Gianduja extends Chocolate, a previously published design method. It does so by providing the robots with the capability to communicate using one message. The semantics of the message is not a priori fixed. It is the automatic design process that implicitly defines it, on a per-mission basis, by prescribing the conditions under which the message is sent by a robot and how the receiving peers react to it. We empirically study Gianduja on three missions and we compare it with the aforementioned Chocolate and with EvoCom, a rather standard evolutionary robotics method that generates communication-based behaviors. We evaluate the behaviors produced by the three automatic design methods on a swarm of 20 e-puck robots. The results show that Gianduja uses communication meaningfully and effectively in all the three missions considered. The aggregate results indicate that, on the three missions considered, Gianduja performs significantly better than the two other methods under analysis.

1 Introduction

In swarm robotics, communication plays a central role and can significantly enhance collective performance [3]. Designing effective communication mechanisms is challenging and design choices can have an important impact on the effectiveness, complexity, and cost of a swarm [2]. Notwithstanding the advancements achieved in the last decade [4,7,24,29,34,43,51], the design of robot swarms is still at dawn and no generally applicable methodology has been proposed so far [8,11,21]. Automatic design methods are a promising way of approaching the issue [6,15]. In automatic methods, the design problem is cast into an optimization problem: a space of solutions is searched via an optimization algorithm, with the goal of maximizing a performance measure. Most of the

The proposed method was implemented and tested by KH. The experiments were designed by the three authors. This paper was drafted by KH, refined by MB, and revised by the three authors. The research was conceived and directed by MB.

M. Dorigo et al. (Eds.): ANTS 2018, LNCS 11172, pp. 16–29, 2018.
https://doi.org/10.1007/978-3-030-00533-7_2

research on the automatic design of robot swarms has been inspired by neuro-evolution [37,47]. In this approach, robots are controlled by a neural network, whose parameters are obtained via artificial evolution [12,27,31,38,44,45,47,50]. Other methods have been proposed that are based on different control architectures and/or different optimization algorithms [16,18,22,30]. Among them, `Chocolate` [16] produces probabilistic finite state machines by using the irace optimization algorithm [35] to assemble preexisting low-level behaviors and conditions, and to fine-tune their parameters. The low-level behaviors, which define the actions that individual robots can perform, are: exploration, stop, phototaxis, anti-phototaxis, attraction to neighbors, repulsion from neighbors. The conditions, which define events that cause a transition between low-level behaviors, are: black-floor, white-floor, gray-floor, neighbor-count, inverted-neighbor-count, fixed-probability.

In this paper, we study the automatic design of collective behaviors that rely on communication. In particular, we are interested in exploring the case in which messages exchanged by the robots do not have an a priori defined semantics. We wish to develop an automatic design process that, on a per-mission basis, defines (i) the conditions under which a robot broadcasts a message and (ii) the effects that this message has on the behavior of the receiving peers.

We introduce `Gianduja`, a new instance of AutoMoDe [18]. `Gianduja` extends `Chocolate` by adding the capability of locally broadcasting a single message and reacting to it. We test `Gianduja` on three missions that we shall call AGGREGATION, STOP, and DECISION. We present results of experiments performed with a swarm of 20 e-puck robots [36].

Within the evolutionary robotics approach, it has already been shown that an automatic design process can (implicitly) give a semantics to an a priori meaningless message. Nonetheless, this has been demonstrated only on teams of two robots [1,49]. The novel elements that we propose in this paper are that: (1) we study the emergence of a message semantics in swarm robotics and we demonstrate it with a swarm of 20 robots; (2) we show that a message semantics can emerge also when robots are controlled by a finite state machine; and (3) we consider three different missions in which the emerging semantics is different.

2 Related Work

Communication—be it direct or indirect, explicit or implicit—is an integral part of most robot swarms demonstrated so far. As a result, the literature on communication in swarm robotics is extremely large and covering it goes beyond the scope of this paper. In particular, we will not cover studies in which communication has been a priori defined by the designer—e.g., [2,3,9,14,28]. Instead, we will focus on studies in which communication has been automatically designed.

The vast majority of studies in which communication emerged from an automatic design process belong within evolutionary robotics [37,47,48]. Quinn et al. [41,42] were the first to study the emergence of communication between agents. In their studies, robots move in an arbitrary direction while staying close to each other. Robots do not have dedicated communication devices. Nonetheless, they evolved a simple form of implicit communication: using their proximity sensors, robots detect motion in their peers and establish a social interaction. In particular, they coordinate to assume the roles of leader or follower. Nolfi [39] evolved a behavior for solving a collective navigation problem. Robots are controlled by neural networks and can communicate using four different signals. Although the evolutionary process did not explicitly reward the use of communication, it produced a behavior in which the robots effectively use communication to coordinate. The behavior obtained was tested in simulation on a swarm of four robots. Floreano et al. [13] studied the evolution of robots that can produce visual signals to provide information on food location. The authors evolved behaviors for a swarm of ten robots that were eventually able to reliably find the food source. Communication increases the performance of the swarm compared to the case in which robot cannot communicate. The behavior was then tested with real robots. Ampatzis et al. [1] evolved the behavior of two robots to recognize features of the environment and react accordingly. The robots are controlled by neural networks and can use their on-board speakers and microphones to send/receive a sound message. Although communication is not strictly needed to solve the task and was not explicitly rewarded in the evolutionary process, it emerged as it improves performance. The behavior obtained was tested both in simulation and in reality with two s-bot robots. Tuci [49] studied the origin of communication from an evolutionary perspective. The author considered a setting in which two robots, which might communicate via a sound message, need to categorize the environment and act accordingly. Also in this case, although communication was not explicitly rewarded, the evolutionary process produced behaviors that effectively use the available communication capabilities to perform the mission. Experiments were conducted in simulation only.

Among all the studies highlighted above, the research we present in this paper is most closely related to [1,49]. Indeed, as in those studies, we consider the case in which the semantics of the message exchanged by the robots is not a priori defined but is the result of the automatic design process.

3 AutoMoDe-`Gianduja`

By introducing `Gianduja`, we address one of the limitations of AutoMoDe: the instances of AutoMoDe defined so far, `Vanilla` and `Chocolate`, are unable to design behaviors that exploit explicit communication. The behaviors automatically generated by `Gianduja` can rely on sending and receiving a single message whose semantics is not fixed a priori. `Gianduja` is a proper extension of `Chocolate` [16] that adds the ability to (i) locally broadcast a message, (ii) change state when the message is received (or is not received), and (iii) approach (or retract from) neighboring peers that broadcast the message.

As `Chocolate` and `Vanilla`, `Gianduja` designs control software for the e-puck platform. Nonetheless, it considers a reference model that is an extension of the one considered by `Chocolate` and `Vanilla`—RM 1.1 [25]. Precisely, the

Table 1. Reference model RM 2: novelties with respect to RM 1.1 are highlighted.

Input	Value	Description
$prox_{i\in\{1,\dots,8\}}$	[0,1]	reading of proximity sensor i
$light_{i\in\{1,\dots,8\}}$	[0,1]	reading of light sensor i
$gnd_{j\in\{1,2,3\}}$	{black, gray, white}	reading of ground sensor j
n	[0,20]	number of neighboring robots perceived
V	$\big([0,0.70]\,\mathrm{m},[0,2\pi]\,\mathrm{rad}\big)$	direction of attraction to them
b	[0,20]	number of messaging neighbors perceived
V_b	$\big([0,0.70]\,\mathrm{m},[0,2\pi]\,\mathrm{rad}\big)$	direction of attraction to them

Output	Value	Description
$v_{k\in\{l,r\}}$	$[-0.12,0.12]\,\mathrm{m\,s^{-1}}$	target linear wheel velocity
s	$\{on,\ off\}$	broadcast message

Period of the control cycle: 100 ms

extension concerns the ability to (a) locally broadcast the message and (b) sense the broadcasting peers that are within the perception range. The new reference model, which we shall call RM 2, is given in Table 1. The variables highlighted are the elements of novelty with respect to RM 1.1: b, V_b, and s. Before we explain these variables, it is convenient that we first recall the mechanism that allows robots to perceive their neighbors—both in RM 1.1 and in RM 2. Using their range-and-bearing module [23], all robots continuously broadcast a "heartbeat" signal whose payload encodes their unique ID. At every time step, every robot receives the heartbeat signal of the peers that are within its perception range, which is of about 0.70 m. It can therefore infer the number of neighboring peers and their relative positions: range and bearing. This information is made available to the control software via the variables r and V. The former is the number of neighboring peers and the latter is a vector indicating the direction of attraction to these neighboring peers, which is computed based on the framework on virtual potential fields [46].

In RM 2, every robot locally broadcasts the message by setting a specific bit of its heartbeat's payload. Due to this extension, at every time step, a robot can infer the number and relative position of the neighboring peers that are broadcasting the message. The information that is made available to the control software is stored in the variables b and V_b. The former is the number of neighboring peers that broadcast the message and the latter is a vector indicating the direction of attraction to these neighboring peers, which also in this case is computed following the framework on virtual potential fields [46]. Formally,

$$V_b = \begin{cases} \sum_{m=1}^{b} (\alpha/r_m^2, \angle b_m), & \text{if } b > 0 \text{ broadcasting robots are perceived;} \\ (1, \angle 0), & \text{otherwise.} \end{cases}$$

Here, r_m and $\angle b_m$ are the range and bearing of the m-th neighboring peer that is broadcasting the message and α a real value parameter. The variable s can be set by the control software and indicates whether, during the following control cycle, the robot should broadcast the message or not. It can take two values: *on* or *off*.

`Gianduja` produces control software in the form of probabilistic finite state machines, as `Chocolate` does. It does so by combining and fine-tuning (a) the original transition conditions of `Chocolate` (and `Vanilla`) [16,18]; (b) an extended version of the low-level behaviors of `Chocolate` (and `Vanilla`) [16,18]; (c) four additional modules: two low-level behaviors and two transition conditions. We extend the preexisting low-level behaviors of `Chocolate` (and `Vanilla`) by adding a binary parameter: if the parameter is set, the robot continuously broadcasts the message while performing the low-level behavior; otherwise, it does not. We conceived the four additional modules specifically for exploiting the extended functionalities provided by RM 2. The two additional low-level behaviors are: **attraction to message** – the robot moves in the direction indicated by V_b; **repulsion from message** – the robot moves in the opposite direction. Also these additional behaviors have the aforementioned binary parameter that specifies whether the message should be broadcast or not. The two additional conditions are: **message count** – a state transition occurs if the number of neighboring peers broadcasting the message is larger than the value of a parameter; **inverted message count** – a state transition occurs if the number of neighboring peers broadcasting the message is smaller than the value of a parameter. The additional modules are modeled after the original attraction, repulsion, neighbor-count, and inverted-neighbor-count of `Chocolate` (and `Vanilla`) [16,18]. The optimization algorithm used to search the space of the possible probabilistic finite state machines that can be obtained by assembling the available modules and fine-tuning their parameters is irace [35]—the same algorithm used in `Chocolate`. As in `Chocolate` (and `Vanilla`), valid probabilistic finite state machines have at most four states and each state has at most four outgoing transitions. Finally, as in `Chocolate` (and `Vanilla`), the design process is performed in simulation using ARGoS [20,40].

4 Experimental Setting

We test `Gianduja` on three missions and we compare it with two other methods.

4.1 Missions

In all three missions, the robots operate in a dodecagonal area of $4.91\,\mathrm{m}^2$. The arena is surrounded by walls. Its floor is gray, apart from some specific areas that, on a per-mission basis, could be white or black, as detailed in the following. The time available to the robots for performing a mission is $T = 120\,\mathrm{s}$. The three missions considered are AGGREGATION, STOP, and DECISION; they are described in the following. We have selected them because, according to our a priori expectations, communication should play a different role in them. Indeed, we expect that AGGREGATION can be solved without using communication. On the other hand, we expect that STOP and DECISION require communication for being solved effectively. We also expect that the semantics implicitly attached to the message by the automatic design process will be different in STOP and DECISION. We will detail this in the following, on a per-mission basis.

AGGREGATION. The arena's floor is marked by two circular spots, with diameter of 0.6 m: one is white and the other black. They are positioned on the left-hand side of the arena, separated by a gap of 0.25 m. At the beginning of each run, the robots are randomly positioned in the right-hand half of the arena, so that no robot is already on the spots—see Fig. 1(*right*). The mission prescribes that the robots quickly aggregate on the white spot. The black spot is not supposed to play any role and simply acts as a disturbance to the automatic design process. The performance of the robots is measured via the following objective function—the higher, the better:

$$C_{\text{A}} = 24000 - \sum_{t=1}^{T}\sum_{i=1}^{N} I_i(t); \qquad I_i(t) = \begin{cases} 0, & \text{if robot } i \text{ is on the white spot;} \\ 1, & \text{otherwise.} \end{cases}$$

Here, i is an index that spans over all the robots of the swarm, N is the total number of robots, and $T = 120\,\text{s}$ is duration of the experiment. 24 000 is the maximum theoretical score that the robots could achieve. It is included in the definition of the objective function to guarantee that its value is non-negative and ranges from 0 to its theoretical maximum.

As already mentioned, we think communication is not needed in this mission.

STOP. The arena's floor is marked by a circular white spot, with diameter of 0.2 m, positioned near the walls, on the top-left quadrant. At the beginning of each experimental run, all robots are randomly positioned in the right-hand half of the arena: none of them is on the white spot—see Fig. 1(*center*). The mission prescribes that the robots search for the spot and, as one of them finds it, all stop quickly. The performance measure—the higher, the better—is:

$$C_{\text{S}} = 48000 - \left(\bar{t}N + \sum_{t=1}^{\bar{t}}\sum_{i=1}^{N} \bar{I}_i(t) + \sum_{t=\bar{t}+1}^{T}\sum_{i=1}^{N} I_i(t) \right);$$

$$I_i(t) = \begin{cases} 1, & \text{if robot } i \text{ is moving;} \\ 0, & \text{otherwise;} \end{cases} \qquad \bar{I}_i(t) = 1 - I_i(t).$$

Here, i, N, and T are defined as above; $\bar{t}$ is the time at which a robot steps on the white spot for the first time. The performance measure ranges from 0 to its maximum of 48 000. In the definition of I_i (and $\bar{I}_i$), a robot is considered to be moving if its center has traveled more than 5 mm in the last time step.

We expect that communication is needed in this mission and that `Gianduja` produces behaviors in which (i) robots broadcast the message if they step on the white spot; (ii) upon receiving the message, robots stop and possibly relay it.

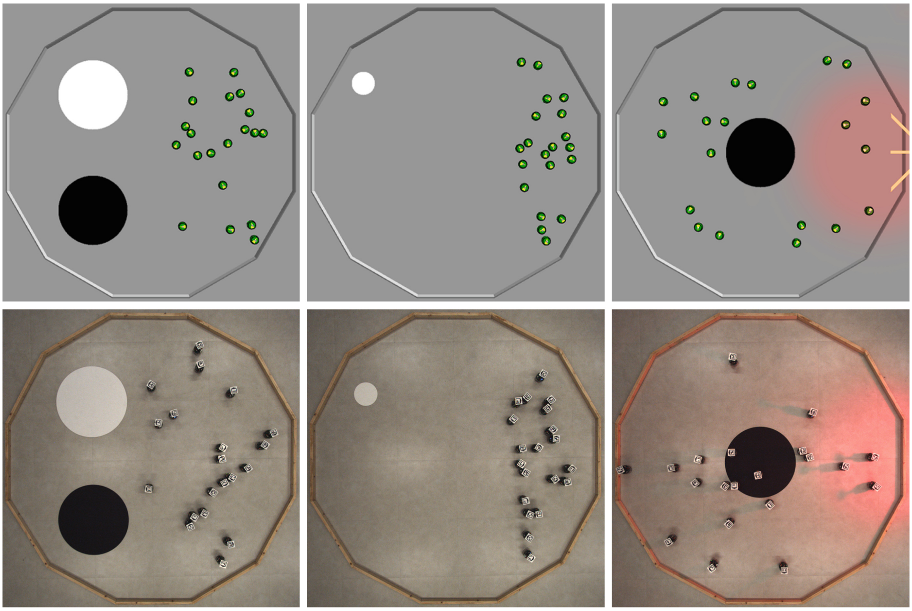

Fig. 1. Arenas for the three missions: AGGREGATION, STOP, and DECISION (*from left to right*); simulation (*top*) and real setup (*bottom*).

DECISION. The arena's floor is marked by a circular spot, with diameter of 0.6m, located in the center of the arena. The spot can be either white or black, with a probability of 0.5. A light source is placed outside the arena, on the right-hand side. At the beginning of each run, robots are randomly positioned—see Fig. 1(*right*). The mission prescribes that the robots quickly relocate into the right-hand half of the arena, when the spot is black; and into the left-hand half, when the spot is white. The performance measure—the higher, the better—is:

$$C_{\text{D}} = 24000 - \sum_{t=1}^{T} \sum_{i=1}^{N} I_i(t);$$

$$I_i(t) = \begin{cases} 0, & \text{if robot } i \text{ is in the correct half of the arena;} \\ 1, & \text{otherwise.} \end{cases}$$

Here, i, N, and T are defined as above. The performance measure ranges between 0 and its theoretical maximum of 24 000.

Also in this case, we expect that communication is needed. A straightforward solution would require two distinct messages: one per spot color/half of the arena in which robots should relocate. As the robots have only one message available, the solution we foresee is that they go in one direction by default and revert to the opposite one in case they receive a message sent by a robot that steps on the spot, should its color indicate that the correct direction is not the default one.

4.2 Protocol

We compare `Gianduja` with `Chocolate` [16,32] and `EvoCom`. `Chocolate` was originally defined in [16] and is used here unmodified. `EvoCom` is an evolutionary method that we introduce for this study. It is an extension of `EvoStick`—a design method that, via an evolutionary process, tunes the parameters of a neural networks to control the e-puck platform, as it is modeled by RM 1.1. `EvoStick` was formally defined in [18] to serve as a yardstick in the study of `Vanilla`, but had been previously analyzed in [19]. It was subsequently included in other empirical studies [5,16,17,33]. `EvoCom` targets the e-puck platform, as it is modeled by RM 2—see Table 1. With respect to `EvoStick`, it has the further capability of locally broadcasting a message and reacting to it. It features (i) one extra output node for s; and (ii) five extra input nodes: one for b and four for the projections of V_b on the four unit vectors pointing at 45°, 135°, 225°, and 315° with respect to the head of the robot. The neural network is optimized using a standard evolutionary algorithm, the same adopted in `EvoStick`—see [18,32] for the details. Artificial evolution is based on simulations performed with ARGoS [20,40]—under the same conditions that hold for `Gianduja` and `Chocolate`.

We consider a swarm of 20 e-puck robots. For each of the three missions, each of the three methods under analysis is executed 15 times to obtain 15 instances of control software. Each design process can rely on a maximum of 200 000 simulated runs. The simulator adopted in the study is ARGoS3, beta 48. We evaluate each instance of control software obtained by the three design methods: once in simulation and once on the physical robots. The initial positions of the robots and the order of the experimental runs are randomized to avoid any bias. In robot experiments, the value of the objective function is computed automatically using a tracking system that extracts information from images taken with an overhead camera every 100 ms.

Statistics. We report per-mission boxplots of the performance registered in simulation and reality. When appropriate, we report also the outcome of a Wilcoxon rank-sum test, at 95% confidence [10]. Eventually, we aggregate all the results of the robot experiments by ranking across each mission the performance obtained by the instances of control software generated by each method. We present the outcome of a Friedman test [10] in a plot that displays the average rank of each method and its 95% interval of confidence. If two intervals do not overlap, the results we registered for the corresponding methods are significantly different. In the following, statements like "A performs significantly better that B" imply that an appropriate statistical test—either a Wilcoxon or a Friedman test—has been employed and has detected significance with confidence of at least 95%.

5 Results

We present the results on a per-mission basis and then we aggregate them across the three missions. Numerical results, videos, code, and finite state machines generated by `Gianduja` and `Chocolate` are available in [26].

AGGREGATION. Results are reported in Fig. 2(*left*). Both Gianduja and Chocolate perform significantly better than EvoCom. Although Gianduja performs significantly better than Chocolate in simulation, the results of the two methods on the robots are similar.

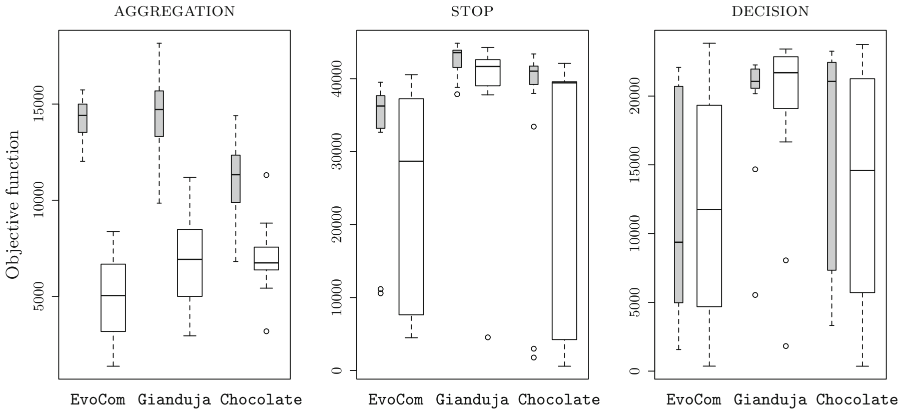

Fig. 2. AGGREGATION, STOP, and DECISION (*from left to right*). Thick white boxes represent the results of robot experiments; thin gray ones, those of simulations.

At visual inspection, EvoCom seems to be unable to use communication effectively. The robots randomly explore the arena—sometimes forming moving clusters. If they enter the white spot, they spin in place. Also in Chocolate and Gianduja, robots navigate randomly, but they stop upon reaching the white spot. In Gianduja, when on the white spot, robots typically broadcast the message; the receiving peers converge towards them eventually reaching the white spot. Although elegant, Gianduja's solution does not significantly improve over Chocolate's one. It can be observed that Gianduja suffers the reality gap more than Chocolate. This could be due to the fact that the ground sensor of the e-puck robot is quite prone to report false positives in the detection of white/black floor. As it can be seen in simulation, the behaviors produced by Gianduja rely on communication to attract peers once the white spot is detected. In the presence of false positives, this feature could hinder performance. Chocolate, which does not rely on communication, is apparently less affected by false positives.

STOP. Results are reported in Fig. 2(*center*). Gianduja performs significantly better than EvoCom and Chocolate, both in simulation and reality.

At visual inspection, EvoCom seems to be unable to use its communication capabilities effectively, whereas Gianduja does. In Gianduja, robots move randomly until one reaches the white spot and stop. This robot broadcasts the message. The receiving peers relay it and stop. In EvoCom and Chocolate (which is not endowed with communication capabilities), robot move in random directions until being stopped by the walls. Although trivial, this behavior often scores

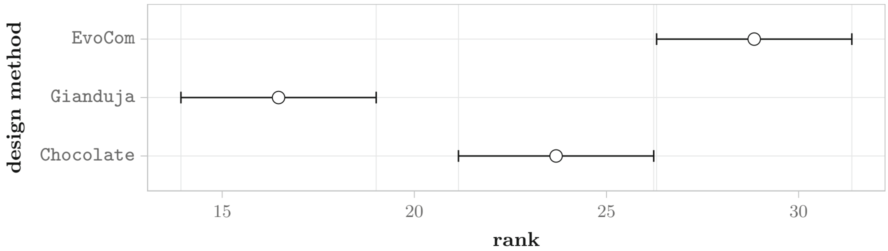

Fig. 3. Friedman test on the aggregated results of the three missions. The plot represents the average rank of the three methods and their 95% confidence interval.

better than one would expect because there is some relatively high chance that, before robots stop against a wall, at least one of them has reached the white spot. In this mission, the behaviors generated by `Gianduja` cross the reality gap better than those of `EvoCom` and `Chocolate`.

DECISION. Results are reported in Fig. 2(*right*). `Gianduja` performs significantly better than `EvoCom` both in simulation and reality. Concerning the comparison between `Gianduja` and `Chocolate`, although the difference is not significant in simulation, on the robots `Gianduja` performs significantly better.

`Gianduja` uses communication effectively. By default, robots go towards one side of the arena. If one robot steps on the central spot and its color indicates that the correct side of the arena is not the default one, the robot itself broadcasts the message. Receiving peers relay the message and all robots head to the correct direction. In some instances of control software designed by `Gianduja`, the selected default side is the right-hand one, and in others is the left-hand. Accordingly, robots start by performing phototaxis or anti-phototaxis and then possibly switch depending on the color of the central spot. In `Chocolate`, the behavior is similar but, as the robots are not endowed with communication capabilities, only the robots that individually step on the central spot are able to revert their default choice, should it be needed. `EvoCom` failed to produce any consistently meaningful behavior. The score has a very large variability and appears to be determined by chance. In simulation, the performance observed is even worse than random behavior, which should produce an expected score of 12 000—half of the maximum. In robot experiments, the score observed matches the profile of a random behavior. `Gianduja`'s behaviors cross the reality gap nicely while those of `Chocolate` appear to experience a large performance drop. As the performance of `EvoCom` is particularly poor, any consideration on how the method handles the reality gap would be meaningless.

Aggregate Results. The aggregate results are presented in Fig. 3. The plot confirms that, across the three missions considered, `Gianduja` performs significantly better that both `Chocolate` and `EvoCom`.

6 Conclusions

We have studied the problem of the automatic design of collective behaviors that rely on communication. We have focused on the case in which robots are able to locally broadcast a single message whose semantics is not fixed a priori: the automatic design method can re-define it on a per-mission basis, as needed.

We have introduced `Gianduja`, an automatic design method based on the previously published `Chocolate`. `Gianduja` generates control software by assembling preexisting software modules into a probabilistic finite state machine. We tested `Gianduja` on three missions, showing that the way in which the message is used by the robots is different—and meaningful—in each of them. As desired, the (implicit) semantics of the message is automatically defined on a per-mission basis by the design process. On all three missions, `Gianduja` performs significantly better that `EvoCom`, a rather standard evolutionary robotics methods for robots that are able to broadcast and receive a message. On two of the three missions, `Gianduja` performs also significantly better than `Chocolate`, which is not endowed with communication capabilities. The only mission on which the performance of `Gianduja` and `Chocolate` is comparable is one in which we a priori expected that communication is not needed. When aggregated, the results of the robot experiments indicate that, across the three missions considered, `Gianduja` performs significantly better that both `EvoCom` and `Chocolate`.

On the missions considered, `Gianduja` has also shown a weakness: It appears to be more sensitive than `Chocolate` to noisy readings from the ground sensor. We observed this issue in AGGREGATION but it might have had an impact also in the other two missions. The reason why this issue has a relative lower impact in the other missions is possibly that communication is strictly needed to accomplish them. This clearly gives a major advantage to `Gianduja` over `Chocolate` and greatly compensates the increased sensitivity to sensor noise.

Future work will focus on testing `Gianduja` on further missions. We will also study the possibility of extending `Gianduja` so that it can handle multiple messages and therefore generate more complex collective behaviors. Finally, we will address also the sensitivity of `Gianduja` to sensor noise. A possible way to handle the issue is to improve the noise models used in simulation so as to produce behaviors that are more robust to false positives in the detection of white/black ground. We are considering also to adopt ideas from game theory to prevent that malicious (or simply fallacious, erroneous, unintended) messages propagate across the swarm and negatively impact its collective behavior.

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). Mauro Birattari acknowledges support from the Belgian *Fonds de la Recherche Scientifique* – FNRS.

References

1. Ampatzis, C., Tuci, E., Trianni, V., Dorigo, M.: Evolution of signaling in a multi-robot system: categorization and communication. Adapt. Behav. **16**(1), 5–26 (2008)
2. Balch, T.: Communication, diversity and learning: cornerstones of swarm behavior. In: Şahin, E., Spears, W.M. (eds.) SR 2004. LNCS, vol. 3342, pp. 21–30. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30552-1_3
3. Balch, T., Arkin, R.C.: Communication in reactive multiagent robotic systems. Auton. Robot. **1**(1), 27–52 (1994)
4. Berman, S., Kumar, V., Nagpal, R.: Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination. In: Zexiang, L. (ed.) IEEE International Conference Robotics and Automation, ICRA, pp. 378–385. IEEE Press, Piscataway (2011)
5. Birattari, M., Delhaisse, B., Francesca, G., Kerdoncuff, Y.: Observing the effects of overdesign in the automatic design of control software for robot swarms. In: Dorigo, M. (ed.) ANTS 2016. LNCS, vol. 9882, pp. 149–160. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44427-7_13
6. Bozhinoski, D., Birattari, M.: Designing control software for robot swarms: software engineering for the development of automatic design methods. In: ACM/IEEE 1st International Workshop on Robotics Software Engineering. RoSE, pp. 33–35. ACM, New York (2018)
7. 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 Trans. Auton. Adapt. Syst. **9**(4), 17.1–17.28 (2015)
8. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. Swarm Intell. **7**(1), 1–41 (2013)
9. Cao, Y., Fukunaga, A., Kahng, A., Meng, F.: Cooperative mobile robotics: antecedents and directions. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, vol. 1, pp. 226–234. IEEE Press, Piscataway (1997)
10. Conover, W.J.: Practical Nonparametric Statistics, 3rd edn. Wiley, New York (1999)
11. Dorigo, M., Birattari, M., Brambilla, M.: Swarm robotics. Scholarpedia **9**(1), 1463 (2014)
12. Floreano, D., Husbands, P., Nolfi, S.: Evolutionary robotics. In: Handbook of Robotics, pp. 1423–1451 (2008)
13. Floreano, D., Mitri, S., Magnenat, S., Keller, L.: Evolutionary conditions for the emergence of communication in robots. Curr. Biol. **17**(6), 514–519 (2007)
14. Fong, T., Nourbakhsh, I.: Socially interactive robots. Robot. Auton. Syst. **42**(3–4), 139–141 (2009)
15. Francesca, G., Birattari, M.: Automatic design of robot swarms: achievements and challenges. Front. Robot. AI **3**(29), 1–9 (2016)
16. Francesca, G., et al.: AutoMoDe-chocolate: automatic design of control software for robot swarms. Swarm Intell. **9**(2/3), 125–152 (2015)
17. Francesca, G., et al.: An experiment in automatic design of robot swarms. In: Dorigo, M. (ed.) ANTS 2014. LNCS, vol. 8667, pp. 25–37. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09952-1_3
18. 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**(2), 89–112 (2014)

19. Francesca, G., Brambilla, M., Trianni, V., Dorigo, M., Birattari, M.: Analysing an evolved robotic behaviour using a biological model of collegial decision making. In: Ziemke, T., Balkenius, C., Hallam, J. (eds.) SAB 2012. LNCS (LNAI), vol. 7426, pp. 381–390. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33093-3_38
20. Garattoni, L., Francesca, G., Brutschy, A., Pinciroli, C., Birattari, M.: Software infrastructure for e-puck (and TAM). Technical report TR/IRIDIA/2015-004, IRIDIA, Université libre de Bruxelles, Belgium (2015)
21. Garattoni, L., Birattari, M.: Swarm robotics. In: Webster, J. (ed.) Wiley Encyclopedia of Electrical and Electronics Engineering. Wiley, Hoboken (2016)
22. Gauci, M., Chen, J., Li, W., Dodd, T.J., Groß, R.: Self-organized aggregation without computation. Int. J. Robot. Res. **33**(8), 1145–1161 (2014)
23. Gutiérrez, Á., Campo, A., Dorigo, M., Donate, J., Monasterio-Huelin, F., Magdalena, L.: Open e-puck range & bearing miniaturized board for local communication in swarm robotics. In: Kosuge, K. (ed.) IEEE International Conference on Robotics and Automation, ICRA, pp. 3111–3116. IEEE Press, Piscataway (2009)
24. Hamann, H., Wörn, H.: A framework of space-time continuous models for algorithm design in swarm robotics. Swarm Intell. **2**(2), 209–239 (2008)
25. Hasselmann, K., Ligot, A., Francesca, G., Birattari, M.: Reference models for AutoMoDe. Technical report TR/IRIDIA/2018-002, IRIDIA, Université libre de Bruxelles, Belgium (2018)
26. Hasselmann, K., Robert, F., Birattari, M.: Automatic design of communication-based behaviors for robot swarms: supplementary material. http://iridia.ulb.ac.be/supp/IridiaSupp2018-003/ (2018)
27. Jakobi, N., Husbands, P., Harvey, I.: Noise and the reality gap: the use of simulation in evolutionary robotics. In: Morán, F., Moreno, A., Merelo, J.J., Chacón, P. (eds.) ECAL 1995. LNCS, vol. 929, pp. 704–720. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-59496-5_337
28. Jones, C., Mataric, M.J.: Automatic synthesis of communication-based coordinated multi-robot systems. In: International Conference on Intelligent Robots and Systems, IROS, vol. 1, pp. 381–387. IEEE Press, Piscataway (2004)
29. Kazadi, S., Lee, J.R., Lee, J.: Model independence in swarm robotics. Int. J. Intell. Comput. Cybern. **2**(4), 672–694 (2009)
30. König, L., Mostaghim, S.: Decentralized evolution of robotic behavior using finite state machines. Int. J. Intell. Comput. Cybern. **2**(4), 695–723 (2009)
31. Koos, S., Mouret, J.B., Doncieux, S.: The transferability approach: crossing the reality gap in evolutionary robotics. IEEE Trans. Evol. Comput. **17**(1), 122–145 (2013)
32. Ligot, A., Hasselmann, K., Delhaisse, B., Garattoni, L., Francesca, G., Birattari, M.: AutoMoDe, NEAT, and EvoStick: implementations for the e-puck robot in ARGoS3. Technical report TR/IRIDIA/2017-002, IRIDIA, Université libre de Bruxelles, Belgium (2017)
33. Ligot, A., Birattari, M.: On mimicking the effects of the reality gap with simulation only experiments. In: Dorigo, M. (ed.) ANTS 2018. LNCS, vol. 11172, pp. 109–122. Springer, Berlin (2018)
34. Lopes, Y.K., Trenkwalder, S.M., Leal, A.B., Dodd, T.J., Groß, R.: Supervisory control theory applied to swarm robotics. Swarm Intell. **10**(1), 65–97 (2016)
35. López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., Stützle, T.: The irace package: iterated racing for automatic algorithm configuration. Oper. Res. Perspect. **3**, 43–58 (2016)

36. Mondada, F., et al.: The e-puck, a robot designed for education in engineering. In: Gonçalves, P., Torres, P., Alves, C. (eds.) Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, pp. 59–65. Instituto Politécnico de Castelo Branco, Castelo Branco (2009)
37. Nolfi, S., Floreano, D.: Evolutionary Robotics. MIT Press, Cambridge (2000)
38. Nolfi, S., Floreano, D., Miglino, G., Mondada, F.: How to evolve autonomous robots: different approaches in evolutionary robotics. In: Brooks, R.A., Maes, P. (eds.) Artificial Life IV: Proceedings of the Workshop on the Synthesis and Simulation of Living Systems. pp. 190–197. MIT Press, Cambridge (1994)
39. Nolfi, S.: Emergence of communication in embodied agents: co-adapting communicative and non-communicative behaviours. Connect. Sci. **17**(3–4), 231–248 (2005)
40. Pinciroli, C., et al.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intell. **6**(4), 271–295 (2012)
41. Quinn, M.: Evolving communication without dedicated communication channels. In: Kelemen, J., Sosík, P. (eds.) ECAL 2001. LNCS (LNAI), vol. 2159, pp. 357–366. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44811-X_38
42. 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. Lond. A: Math. Phys. Eng. Sci. **361**(1811), 2321–2343 (2003)
43. Reina, A., Valentini, G., Fernàndez-Oto, C., Dorigo, M., Trianni, V.: A design pattern for decentralised decision making. PLoS One **10**(10), e0140950 (2015)
44. Silva, F., Duarte, M., Correia, L., Oliveira, S., Christensen, A.: Open issues in evolutionary robotics. Evol. Comput. **24**(2), 205–236 (2016)
45. Silva, F., Urbano, P., Correia, L., Christensen, A.L.: odNEAT: an algorithm for decentralised online evolution of robotic controllers. Evol. Comput. **23**(3), 421–449 (2015)
46. Spears, W.M., Spears, D., Hamann, J.C., Heil, R.: Distributed, physics-based control of swarms of vehicles. Auton. Robot. **17**, 137–162 (2004)
47. Trianni, V.: Evolutionary Swarm Robotics. Springer, Berlin (2008)
48. Trianni, V.: Evolutionary robotics: model or design? Front. Robot. AI **1**(13), 1–6 (2014)
49. Tuci, E.: An investigation of the evolutionary origin of reciprocal communication using simulated autonomous agents. Biol. Cybern. **101**(3), 183–199 (2009)
50. Urzelai, J., Floreano, D.: Evolutionary robotics: coping with environmental change. In: Whitney, L.D., et al. (eds.) Proceedings of Conference on the Genetic and Evolutionary Computation Conference, GECCO, pp. 941–948. Morgan Kaufmann, San Francisco (2000)
51. Werfel, J., Petersen, K., Nagpal, R.: Designing collective behavior in a termite-inspired robot construction team. Science **343**(6172), 754–758 (2014)