Applied Soft Computing 113 (2021) 108010

Contents lists available at ScienceDirect

Applied Soft Computing

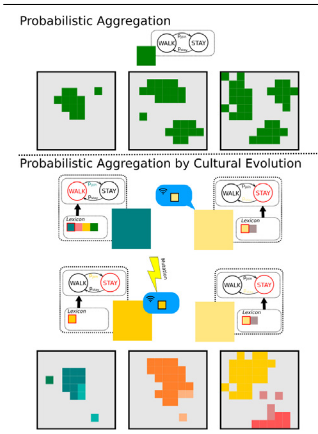
journal homepage: www.elsevier.com/locate/asoc

Cultural evolution of probabilistic aggregation in synthetic swarms

Nicolas Cambier [a,*], Dario Albani [b], Vincent Frémont [c], Vito Trianni [d], Eliseo Ferrante [a,b]

[a] Computational Intelligence Group, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
[b] Technology Innovation Institute, Abu Dhabi, United Arab Emirates
[c] UMR CNRS 6004 LS2N, Ecole Centrale de Nantes, Nantes, France
[d] ISTC, Italian National Research Council, Rome, Italy

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 14 January 2021
Received in revised form 20 August 2021
Accepted 13 October 2021
Available online 29 October 2021

Keywords:
Swarm robotics
Self-organized aggregation
Cultural evolution
Language games

ABSTRACT

Local interactions and communication are key features in swarm robotics, but they are most often fixed at design time, limiting flexibility and causing a stiff and inefficient response to changing environments. Motivated by the need for higher adaptation abilities, we propose that information about emergent collective structures should percolate onto the individual behavior, modifying it in a way that determines suitable responses in the face of new working conditions and organizational challenges. Indeed, complex societies are driven by an evolving set of individual and social norms subject to cultural propagation, which contribute to determining the individual behaviors. We leverage ideas from the evolution of natural language – an undoubtedly efficient cultural trait – and exploit the resulting social dynamics to select and propagate microscopic behavioral parameters that adapt continuously to macroscopic conditions, which in turn affect the agents' communication topography, and, therefore, feed back onto the social dynamics. This concept is demonstrated on a self-organized aggregation behavior, which is a building block for most swarm robotics behaviors and a striking example of how collective dynamics are sensitive to experimental parameters. By means of experiments with simulated and real robots, we show that the cultural evolution of aggregation rules outperforms conventional approaches in terms of adaptivity to multiple experimental settings.

1. Introduction

Using only relatively simple rules and local interactions, swarms of insects are able to collectively build elaborate structures [1,2] and to make qualitative choices in a way that

* Corresponding author.
E-mail addresses: n.p.a.cambier@vu.nl (N. Cambier), dario.albani@tii.ae (D. Albani), vincent.fremont@ec-nantes.fr (V. Frémont), vito.trianni@istc.cnr.it (V. Trianni), e.ferrante@vu.nl (E. Ferrante).

https://doi.org/10.1016/j.asoc.2021.108010

resembles a single complex brain [3,4]. This is self-organization: collective behaviors spontaneously emerging from local interactions, without any central authority dictating it. This phenomenon is ubiquitous in nature and observable in the physical as well as the biological realm [5]. However, biological systems can display a much larger variety of spatio-temporal patterns than purely physical system, thanks to the information processing abilities that characterize the individual agents [6]. In other words, self-organization in biological systems is not solely determined by physico-chemical laws, but can instead count on a wide variety of actions and reactions as determined by the behavior of the numerous (minimally) cognitive agents forming the system. Most importantly, these agents can actively adapt their decisions to the macroscopic environmental and social structures they contribute to forge. For instance, ants use social estimators (e.g., rate of encounters) to adapt their behavior to the actual presence of other individuals and, thus, to the collective response required by the current social context [6]. Such feedback from macroscopic features to microscopic individual behavior is a distinctive quality of biological self-organization. However, decentralized artificial systems like robot swarms mostly overlooked this aspect to date.

Inspired by the power of self-organization, swarm robotics grew in the last two decades, allowing groups of robots to tackle complex tasks cooperatively, controlled by fully decentralized algorithms [7]. Swarm robotics systems indeed have the advantage of being composed of distinct and relatively simplistic parts (i.e., individual robots). As a result, faulty parts are easy to replace and do not impede the execution of the task at hand. Swarms also have a huge potential in terms of scalability and flexibility as the connections between parts are dynamic and can therefore be rearranged as needed. The deployment of swarm robotics systems, however, is complex, as the design of a self-organizing behavior has to deal with two description levels in order to obtain a desired macroscopic behavior from the definition of microscopic individual rules [8]. Specifically, the macroscopic level describes the behavior of the collective as a whole and aims at achieving a specific goal, which is defined in relation to the task or the environment (including the social context). On the other hand, the microscopic level describes the behavioral rules of individual robots and how robots interact locally with each other. Conventionally, robot swarms are designed at the micro-level to achieve macro objectives; homogeneous groups of robots perform a fixed individual behavior which provokes the emergence of a collective behavior at the macro-level [9]. Two aspects that are rarely considered in robots' behaviors are heterogeneity [10,11] and adaptivity, i.e., changing the rules executed by the robots to adjust their individual behavior (and, therefore, the overall swarm organization) to a given context [12].

Previous work has only scratched the surface of the possibilities offered by adaptive local rules, and no example is available – to the best of our knowledge – of individual microscopic behaviors that change in response to macroscopic features related to the social context in which the robots operate. Differently from environmental features, the macroscopic social context is not easily accessible by robots with local sensing and communications. Adapting to such context requires that the macroscopic social dynamics have a bearing on the modifications of the microscopic behavioral rules. A germane example is given by social norms, which are effective only when largely shared within a population [13]. Social dynamics in the form of cultural propagation are key to spread the seeds of social norms, which can however flourish only when their application leads to a generalized advantage [14,15]. When this happens, the social dynamics remain resilient against change, unless a more advantageous norm gets established [16].

To translate this principle in the design of swarm robotics behaviors, the macroscopic dynamics displayed by the swarm need to influence the adaptive change of microscopic rules, so that the latter can continue to support self-organization. The bidirectional feedback between macroscopic and microscopic features can provide large adaptivity of the swarm behavior to a variety of different working conditions, because the system can self-stabilize around suitable collective states that fit the current contingencies, and can rapidly re-adapt when such contingencies change.

To demonstrate this concept in practice, we study how the social dynamics enacted by the cultural propagation of conventions can provide an effective way to determine microscopic behavioral rules. This is achieved by exploiting the synergies between the dynamics of evolutionary linguistic models and a self-organized swarm robotics behavior. Specifically, we focus on the Naming Game (NG), a multi-agent game that studies how linguistic or social norms spread through a population [17], and on the problem of adaptability in the context of self-organized aggregation [18,19]. The latter is a crucial building block of swarm robotics whereby agents have to converge to the same geographic area [20], whose most efficient solutions, differently to ours, usually require advanced sensors and actuators, such as a long-range sensor to detect the presence of other robots [21] or a range-and-bearing sensor to measure relative distance and bearing of neighbors in local range [22], which are not available on many swarm robots. In Section 2, we examine previous works in evolutionary linguistics and swarm robotics that relate to this proposition. Based on the theoretical knowledge brought by these studies, we created a model by which social dynamics spread a "meme" (in this case, the micro-controller's parameters) according to principles of cultural evolution. We present this model in Section 3 and describe its implementation on two platforms: an abstract simulator and real kilobots [23]. In Section 4, we show how self-organized aggregation can benefit from adaptive microscopic rules in a variety of experimental conditions, and demonstrate how the proposed approach successfully applies to a swarm of kilobots. Finally, we present our conclusions in Section 5.

2. Related work

Recent efforts have been made to introduce the dynamics of evolutionary linguistics into the field of swarm robotics, which we review in Section 2.1, along with explaining the necessary background. Section 2.2 instead focuses on self-organized aggregation for robotic swarms.

2.1. Social dynamics in swarm robotics

As agents' actions directly influence their peers, behavioral changes in a few agents can quickly spread to an entire population. This diffusion can follow various patterns depending on the social dynamics at play, from simple epidemics to complex contagion processes [24]. Many models have been developed and proposed to study these patterns, often used as a decision-making mechanism in robotic swarms. Such implementations usually follow an "opinion-based" approach, which is useful for best-of-n decision problems [25], wherein robots have to choose between a limited quantity of options, such as foraging at different food sources. In such cases, robots have an internal representation of their individual choice. Then, they can change this opinion by gathering their peers', either when meeting in specific areas [26,27], or anywhere they encounter [28]. Different rules are possible to account for the information obtained from peers, from the k-unanimity rule (majority opinion between the first k robots encountered) to simple majority [27,29] or a random choice (i.e., voter model [30]). These rules can be based on a more-or-less exhaustive list of opinions present in the swarm, leading to random fluctuations in the decision process that resemble a progressive accumulation of evidence [31]. For tasks that require several agents, the decision can be taken according to some quorum in the team [32]. In most cases, the quality of the swarm's collective decision in ensured by a bias that makes better options more likely to be chosen among the set of opinions. This bias can be explicitly encoded in the behavior (e.g., the duration or frequency of advertising is proportional to the quality of the option [26,27,30]), and/or a natural consequence of the quality of the opinion (e.g., robots that take the shortest route are more often at the nest, where decisions are taken [32]). Alternatively to quorum/voting models, which postulate a pre-defined set of known alternatives and extensive phases of opinion-gathering, other models are based on discovering and evaluating the quality of available options, and on recruitment/inhibition dynamics among peers [33,34].

Another approach to social dynamics comes from evolutionary linguistics that is, the study of the cultural evolution of natural languages [35]. In this context, several minimal algorithms have been proposed, that display specific characteristics of natural languages in controlled experimental settings [36]. These algorithms – referred to as language games (LG) – consist of simple one-to-one interactions, do not require a separated 'voting' phase and can handle a large (potentially infinite [37]) set of opinions.

Many LGs have been developed so far, but the first and simplest LG to be introduced was the Naming Game (NG) [17]. In this game, two or more agents interact to decide on a name for topics they observe in a scene, taking the roles of either *speaker* or *hearer*. Though it was originally developed to study how words were linked to meaning, it more generally illustrates how a norm becomes shared among many agents, and it thus informs our understanding of other social, economic and ecological phenomena [38,39], and does not, therefore, require grounded topics.

The word spoken by the speaker can be chosen stochastically, with probability proportional to an association score that is updated depending on the game's success or failure. Alternatively, in the Minimal Naming Game (MNG), the speaker and hearer can both drop all words associated to the topic (except from the last one they received), in case of success [40]. The MNG has proved to be a valuable model of social dynamics, studied with different approaches, including statistical physics [24]. Studies reported that the MNG can be played with several hearers at once as the speaker can just broadcast its chosen word without needing to wait for any answer [41]. Regardless of these variations, the MNG always results in the whole population agreeing on one word for the topic and suffices to show how self-organization can result in a shared and efficient vocabulary without any generational transmission [37].

LGs and swarm robotics share a similar goal of forming structures (either social or physical) through the self-organization of locally interacting agents. These similarities suggest that the fields of evolutionary linguistics and of swarm intelligence could benefit from each other [42]. Seminal work [43] in this direction implemented the MNG in a swarm of kilobots—small inexpensive robots designed for swarm robotics research [23]. This study focused on the dynamics of the MNG when implemented on agents that are mobile and embodied. In this case, robots were just wandering without any specific task but reaching consensus. The major conclusion from this work is that embodied agents playing a MNG experience a reduced strain on their memory (compared to simulated agents) as the communication interference results in a loss of data and, thus, the abortion of a part of the games. Conversely, collisions among robots lead to slower diffusion and the formation of isolated aggregates that do not interact much with each others, leading to slower convergence than with simulated agents. Despite this, the algorithm still makes the swarm converge to a single word.

The MNG can be more explicitly linked to robot behaviors by, for example, creating new words only in some specific areas of the environment [44]. As the dynamics of the MNG are also influenced by the interaction topology [45], when the topology changes following robots movements – as in a foraging swarm – the game dynamics can reflect the conditions in which consensus is reached. This can result in a meaningful relationship between words and swarm behavior. In a earlier study [46], we focused on the interplay between self-organized behaviors and the MNG, forcing agents to succeed in the MNG to be able to aggregate. In specific environmental conditions, agents partition in a controllable quantity of aggregates, each with its own identifier represented by the word selected by the aggregate. This grounded language, in turn, provides a description of the physical and/or social environment, i.e., a macro-description. In the present paper, we exploit this knowledge by moving one step further, which completes an early preliminary study [47]: We use the descriptive capabilities of the MNG to positively influence the aggregation controller of individual agents in a decentralized fashion.

2.2. Self-organized aggregation

There are two classes of aggregation problems. The first considers environments with heterogeneities that can drive aggregation (e.g., shelters) [48]. This is a class of best-of-n problems [25] and it can therefore be solved with approaches that are similar to the 'opinion-based' models we presented in Section 2.1 [49]. In the second class, aggregation is performed in environments wherein no sites or shelters are initially provided, therefore agents may aggregate anywhere in the environment and the set of alternatives becomes potentially infinite. This is called 'self-organized aggregation'. In this paper, we focus only on the latter as this creates a more challenging and dynamic task. In fact, without shelters, both the number of targets (i.e., aggregates) and their quality (i.e., their size) vary during the course of a single experiments. As this study focuses on the feedback from the macro-level to the micro-controllers, the dynamism of aggregation in a homogeneous environment is the better fit to showcase the advantages of the method proposed here.

Probabilistic aggregation, as observed in cockroaches or bees, is a common behavior-based approach [19,50,51], and often rely on a two-state Probabilistic Finite State Machines (PFSM), with one state to join/stay in a local aggregate, the other to explore. Other PFSM models, which are not directly inspired by nature, use three states as, in addition to exploring, they can approach or be repelled by aggregates [18]. A major setback of probabilistic aggregation, however, is that its efficiency is extremely sensitive to the rate with which agents encounter each other [50]. This rate is itself dependent on many factors, both macroscopic (e.g., population size, arena shape and size) and microscopic (speed, communication range, and even random walk implementation [52]). This high sensitivity has been demonstrated experimentally [19,50] and analytically [50] (i.e., through a macroscopic model). As a consequence, it is very hard to predict the outcome of probabilistic aggregation given a specific set of microscopic rules executed by the robots. Probabilistic aggregation therefore requires supervised tuning of micro-model parameters in order to be effective in any specific setting [18,19].

More recently, probabilistic aggregation controllers were formalized as an optimization problem and solved with *Particle Swarm Optimization* [53]. In addition to being scalable and interpretable, the emerging controller was very similar to [18]. Another approach achieved scalable results with deterministically controlled robots equipped with a single one-bit sensor, without using any PFSM but only a simple sensors-actuators mapping [21]. However, a careful parameter tuning was necessary to select the most efficient controller. It is worth noting that this solution only achieved dynamic aggregates, which are fundamentally easier to achieve as separate flocking aggregate can fetch and absorb each other and do not, therefore, require agents to take individual decisions. In contrast, in static aggregates, agents perpetually have to choose between exploiting the current situation (i.e., remaining within an aggregate whose quality can only be evaluated locally) or exploring in order to find a better situation (i.e., a larger aggregate). An incorrect setting of this exploration/exploitation trade-off fundamentally leads to either free agents that rarely aggregate, or strong small clusters that never disband or merge.

Automatic design methods require a fair amount of offline tuning, whether by automatic optimization [53–56] or brute-force search [21]. The issue with such off-line tuning approaches it that they require a high-level description of the task at hand, which can fail to encompass every aspect of the problem or lead to large reality gaps [57]. Like probabilistic approaches, they are mostly efficient for extremely specific scenarios and must thus be re-executed for any new setting. However, more recent works showed that an automatic design using recursive fitness functions (which therefore encompasses each neighboring state's score into the score of any state), such as the PageRank algorithm [58], could result in less case-specific designs, as models evolved from a single simulated agents are scalable to any swarm size. Another solution to the reality-gap can be found in embodied evolution models [59], where the neural network controller is evolved in each agent, during the course of the experiment, while some genetic exchanges between agents in close proximity are also possible. Like in "opinion-based" approaches, the intrinsic quality of the opinion (in this case, the genetic code) biases the selection thereof. The rewards used to calculate the fitness of the controller are the quantity of horizontal genetic exchanges, and aggregated agents exchange more as they have more peers in their vicinity [60].

Finally, artificial physics approaches have also been used in the last decade in order to maintain consistent robot formations in a decentralized fashion [22,61]. Notably, Leccese et al. [61] developed a swarm aggregation algorithm which, using attractive/repulsive virtual force laws (defined mathematically), produces a moving aggregate, capable of obstacle avoidance. This algorithm also takes the limitations of current swarm robots into account as it accepts saturated input. Similarly, k-nearest neighbor (K-NN) methods [22] aim to construct an Artificial Viscoelastic Mesh. They require an agent to compute an estimate of the swarm density based on local information and to send it to its neighbors, who use this information to build a weight table. Agents then use this weight table to select their nearest neighbors according to different schemes and compute the vector that will maximize the density of the swarm. This results in a behavior whereby the swarm immediately retracts on itself.

Though the works cited above all made significant contributions to the problem of self-organized aggregation, very few moved within the constraints found in earlier works [19,50], namely robots solely capable of estimating the neighbors count. Nevertheless, these limitations are still relevant to swarm robotics as the field addresses tasks at the microscopic level. By only adding the ability to share a few bytes of information, we are able to produce a swarm robotics approach which addresses the issues highlighted above (i.e., rigid controller and lengthy tuning process). Our proposed approach exploits recent developments in automatic designs (embodied evolution) and social dynamics (MNG) to create an adaptive aggregation behavior. The controller itself is based on a probabilistic aggregation model and therefore achieves static aggregation in an homogeneous environment. These conditions are more challenging for the adaptation as the macroscopic definition of the problem (i.e. where to stop) is continuous and dynamic.

3. Model and implementations

In order to test our proposition, we created Probabilistic Aggregation by Cultural Evolution (PA-CE), a novel algorithm that exploits the dynamics of the hearer-only MNG [41] to tune the parameters of a probabilistic aggregation controller. This model is composed of two components: a simple parametrizable aggregation model (which we also use as a baseline in our experiments), presented in Section 3.1, and the cultural evolution mechanism which parametrizes it, as explained in Section 3.2. In order to systematically study the properties of our model, we developed an abstract but fast and efficient multi-agent simulation environment, which is described in Section 3.3. Our metrics and encoding choices are explained in Section 3.4. Finally, as an acid-test, we implemented PA-CE on kilobots as detailed in Section 3.5.

3.1. Probabilistic aggregation

On a micro-level, our aggregation behavior is based on a regular probabilistic aggregation model. It is represented as a Probabilistic Finite State Machine (PFSM) with two states (STAY/WALK) and transition probabilities in-between those. Consequently, an agent never homes in on local aggregates. Rather, it randomly finds them through exploration. To this quite common model, we added the novelty of parametrizable transition functions to switch between the two states (as opposed to values from a table [50] or from a function with constant parameters [19]), as described by Eq. (1) and (2). The probability p_{Leave} to switch from STAY to WALK is computed as follows:

$$p_{Leave}(n) = e^{-bn} \tag{1}$$

where n is the quantity of neighbors within sensing range, so that larger clusters are more attractive and n gives an estimate of the size thereof. This function is modeled as an exponential decay function where b handles the strength of the dispersion: p_{Leave} becomes steeper when b increases, and thus dispersion weakens. On the other hand, the probability p_{Join} grows with n as follows:

$$p_{Join}(n) = \epsilon + \rho(1 - e^{-an}) \tag{2}$$

In this equation, ϵ is the base join probability for $n = 0$ ($p_{Join}(0) > 0$, otherwise no agent would ever stop). Conversely, ρ 'squeezes' the function so that $p_{Join} \leq 1$. In all our experiments, we set $\epsilon = 1 - \rho$ so that p_{Join} always converges to 1. Finally, a is a parameter that handles the strength of the cohesion. Indeed p_{Join} becomes steeper as a increases. In conventional approaches, parameters such as a, b and ρ are left constant over the course of an experiment and across the whole population. In PA-CE, however, these parameters change according to the rules of the MNG to create heterogeneous and adaptive agents, which act according to the size and topology of their neighborhood.

3.2. Probabilistic aggregation by cultural evolution

In PA-CE, any tuple $\langle a, b, \rho \rangle$ represents the convention, or word, that is culturally propagated by the MNG. Each agent has an individual list of words: its lexicon. In order to communicate these words, an agent can take either of two roles: the *speaker* or the *hearer*. As presented in Section 2.1, there a several variations of the MNG. In PA-CE, we used the rule whereby only the hearer updates its lexicon. Consequently, the rules of the MNG in PA-CE are:

1. The speaker chooses a random word in its lexicon and broadcasts it to any agent in the vicinity.
2. The hearer receives the word.
   - If the word is already in the hearer's lexicon (through previous interactions with other agents of the population), all words are removed and replaced by this single word.
   - If the word is unknown, it is added to the lexicon.

At the beginning of an experiment, each agent's lexicon is initialized with one random word (differently from most NG implementations, where words are only created if a speaker has an empty lexicon), which is culturally propagated by the MNG and exchanged as described in Fig. 1. More explicitly, PA-CE follows four steps:

Speaking Agents in the STAY state continuously transmit a word selected randomly from their lexicon. Agents that are not in the STAY state do not communicate. This ensures that only agents that successfully aggregate can propagate their parameters.

Communication noise Words mutate during communication. This is modeled as probabilistic noise, whereby each bit can flip according to a constant probability [62].

Hearing Before any transition in the PFSM, an agent selects a random word from those it has received from speakers in the last time-step and updates its lexicon with it.

Replacement After the update, the lexicon can contain either one word (the game has been successful) or more than one word (the game has been unsuccessful). If the game has been successful, the aggregation parameters are set according to that word, otherwise according to a random word from the lexicon. The decision to leave an aggregate or to join/stay is then taken with probabilities given by Eq. (1) and (2), which are set with that word.

These four steps of PA-CE, with the associated lexicon, are comparable to the elements of embodied evolution [59], and especially to works focused on implicit fitness [63,64]. Indeed, in cultural evolution, the **lexicon** acts as a genetic pool that gathers "cultural material" (i.e., "memes"). **Speaking** and **hearing** form both sides of the mating operation, which implicitly selects genes in biological evolution. Indeed, as the words received are from agents that successfully aggregated (i.e., stopped), the lexicon should converge towards words that enable aggregation. **Hearing** also acts as a selection operator as it updates the lexicon according to the implicit fitness of the "meme" received. Finally, the **communication noise** acts as the mutation operator and promotes the exploration of new parameters.

Differently from the conventional approach, PA-CE features continuously changing parameters both in time and across the individuals within the swarm, which can get fixated as soon as the system self-stabilizes around the usage of a specific setting, as we discuss in Section 4.

3.3. Simulator

The algorithm presented above required a thorough investigation. For this reason, we implemented an abstract multi-agent simulator, which makes it easy to observe and record data from, and is fast enough to run numerous experiments.

We chose to implement a grid world based model (rather than a continuous space) because it allows to simulate collisions, necessary for modeling aggregation, much more efficiently. In this model, the world is a bounded grid and the agents can move in four directions: up, down, right left (which they select randomly). All agents can communicate with agents at a Chebyshev distance of 1 (i.e., diagonals included) in order to favor packed aggregates. Indeed, the maximum neighborhood size is larger than with a simple Manhattan distance (8 neighbors instead of 4), resulting in a larger set of possible responses.

3.4. Encoding and metrics

To model a discrete but large set of words, we encoded the three PFSM parameters, a, b and ρ, into words as follows. The two main parameters, a, b are constrained to the range [0, 5] and ρ is set in the range [0, 1]. We did not explore the range beyond 5 for a and b as, at this point, p_{Join} and p_{Leave} already converge to, respectively, 1 and 0 for $n = 1$. The three parameters are encoded on 8 bits each, for a total of 24 bits. This ensures a fine enough granularity for our purpose (intermediate values of a parameter are unlikely to yield significantly different behaviors than either end) and enables us to apply simple bit flips for the parameters mutation.

The metrics we selected to study our algorithm depend on the following definition of an aggregate: given a graph $G = (V, E)$, were $v \in V$ is an agent and $e \in E$ pairs agents within communication range, an aggregate is a connected component within the graph with size larger than 1. To evaluate the performances of the aggregation, we used the cluster metric, as defined in [21], which is the ratio between the size of the biggest cluster and the population size N. As the goal of self-organized aggregation is for all agents to gather on the same area, this metrics is the most straightforward measure of success. It shows the proportion of agents that were able to meet the largest consensus. We also used the quantity of aggregates as well as the quantity of free agents (i.e., agents that are not in an aggregate) to characterize the dynamics of the aggregation. For example, when the aggregation is unsuccessful, these metrics clearly indicate the cause of failure: either the agents are dispersing too much (never solidifying into a large aggregate) or they are aggregating too much (forming several aggregates rather than a single one).

As outlined in Section 2.2, in order to ensure the portability of our algorithm to novel and/or microscopic robotic platforms, we are assuming a high-degree of constraints with regards to agents' capacities. We therefore chose to develop a baseline case for probabilistic aggregation which would be based on research addressing a similar setting [19,50]. Additionally, the model described in Section 3.1 allows some parameterization in order to optimize aggregation performance, as it is the very best practice currently in swarm robotics [65]. We optimized parameters ρ, a and b (which remain constant during the whole experiment) using irace [66], a parameter tuning method that implements an iterated racing algorithm, which alternates between sampling new parameters settings and comparing the selected settings against each other, i.e., evaluating each setting across several rounds where, for each round, the problem instance is kept constant. Settings are discarded only if they are performing significantly worse than others. The controller, henceforth referred to as Probabilistic Aggregation Fixed and Optimal (PA-FO), was optimized independently for each setting with a budget of 20000 runs each time.

3.5. Kilobot implementation

As a proof of concept, we also implemented PA-CE on kilobots [23]. This platform is evidently richer than our abstract simulation and presents several challenges. For example, communications are less reliable as packets can be lost due to interference. Moreover, the estimate of the distance between a sender

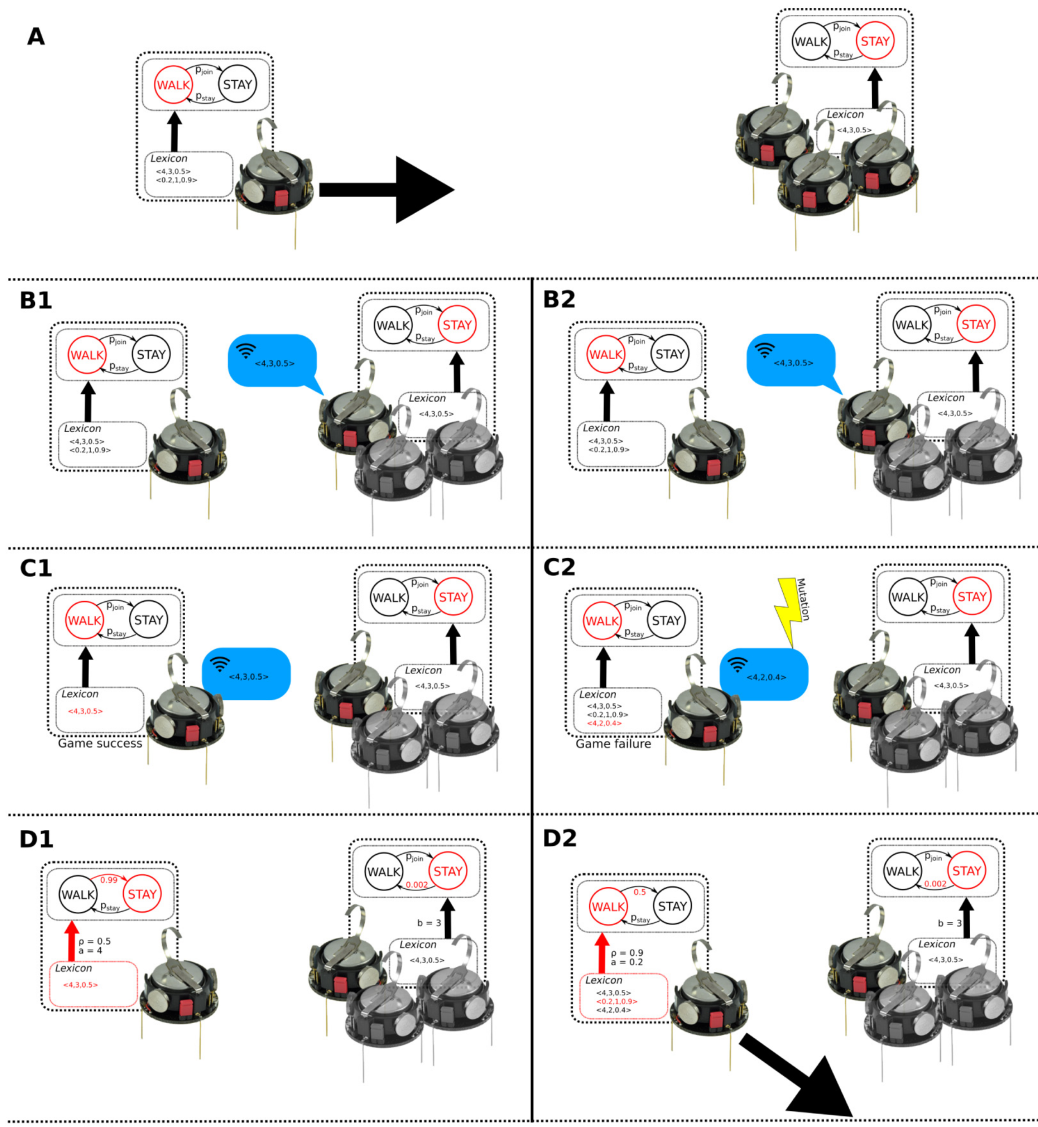


Fig. 1. Illustration of PA-CE. A kilobot (henceforth *hearer*) in the WALK state walks randomly and finds itself near an aggregate **(A)**. Each kilobot in the STAY state are broadcasting a random setting from their lexicon and the arriving kilobot listens to a single one of them (henceforth *speaker*), selected randomly **(B1-2)**. The *hearer* checks the existence of the setting in its lexicon and updates it. In case of game success **(C1)**, the word is maintained in the inventory and all others are excluded. In case of game failure **(C2)**, the word is included into the inventory. Mutations may happen during the transmission of settings, which may cause game failures **(C2)**. Finally, kilobots select a random setting from their lexicon and use this to compute the transition probability to their next state **(D1-2)**. For clarity, only one MNG is illustrated here. However, in PA-CE, every agent (whatever their state) are *hearers* (but only agents in the STAY state are *speakers*).

and a receiver is erratic which, coupled with the packet drops, makes the neighborhood size fluctuate. Physical collisions are also an issue as kilobots can be pushed out of other kilobots' communication range.

In spite of these challenges, we aimed to produce a behavior that would be similar to the one we observed in the grid-world simulations. Therefore, we decided to set a decision time between each consecutive sampling of the probabilities to join or leave, and consequently between each consecutive decisions.

As kilobots walk at a speed of 1 $^{\mathrm{cm}}/_{\mathrm{s}}$ and have a communication range of around 10 cm, we set update time of the PFSM to 10 s, so that the distance walked in a decision step is equal to the communication range, as in the simulations. This, *inter alia*, enables kilobots that decide to leave an aggregate to potentially (if they are going in the right direction) walk out of the latter's communication range before taking a new decision. Whenever a robot is in the STAY state, it continuously emits the same word. The long decision steps (10 s) obviously slow the dynamics down and would prevent the kilobots from reaching full aggregation within their battery lifetime. For this reason, we introduced a slight modification to PA-CE, whereby $p_{Join}(0) = 0.01$ (i.e., an arbitrary low value), whatever the parameter setting. $p_{Join}(n)$ is still given by Eq. (2) for $n > 0$. This ensures that the kilobots rarely stop when they are alone, which speeds the dynamics up, at the cost of creating less opportunities for clusters to form initially.

In order to *hear* during a decision step, a robot only keeps communication messages received within the last 10 time-steps and filters the messages according to their sender's identifier in order to retain only the latest message for each unique neighbor. This is needed due to the asynchronous nature of kilobot communications: It ensures that kilobots receive all messages from their neighbors while discarding duplicates and messages that are too old to be relevant (meaning that the speaker has been out-of-range for a significant time).

For the random walk, the kilobots move in straight lines for 10 s and then draw a new random direction from a uniform distribution in the range $[-\pi, \pi]$.

Table 1
Grid sizes and optimal fixed parameters.

Density	Param.	N=25	N=50	N=100	N=200
High (1/12)	Size	17 × 17	24 × 24	35 × 35	49 × 49
	a	2.86	4.27	0.61	0.18
	b	2.57	2.08	1.8	1.42
	ρ	0.35	0.95	0.22	0.11
Low (1/25)	Size	25 × 25	35 × 35	50 × 50	71 × 71
	a	3.55	2.36	3.4	2.38
	b	2.67	2.1	1.89	1.42
	ρ	0.89	0.7	0.6	0.62

Experiments with kilobots were performed with PA-CE over 10 runs per swarm size ($N = \{25, 50, 100\}$). Due to material reasons, these experiments were all performed in the same 95 cm × 95 cm arena. An overhead camera was set above this arena and it was oriented on the axis orthogonal to the floor of the arena. Experiments were terminated after 1 h (3600 s).

4. Results and discussion

In this section, we present and discuss our results. Section 4.1 is an evaluation of the scalability of PA-CE, compared to PA-FO. Sections 4.2 and 4.3 focus, respectively, on the unique stabilization and dispersion dynamics enabled by PA-CE. Finally, in Section 4.4, we show that our model is efficient in non-abstract situations by studying its scalability when implemented on real kilobots.

4.1. Scalability

As scalability is one of the major problems of self-organized aggregation, we developed an experimental procedure to evaluate how PA-CE deals with varying swarm size, as a measure of the adaptivity of our cultural evolution method. To this end, the simulated experiments were performed with $N = \{25, 50, 100, 200\}$ agents in constant density conditions, to disentangle the effect of the swarm size from the effect of density, as density has been shown to have a major impact in aggregation [46]. To determine separately the effect of density, two densities were considered: high (1 agent every 12 cells) and low (1 agent every 25 cells). The corresponding grid sizes are detailed in Table 1.

To evaluate the performance of our approach, we compared PA-CE to a baseline by the PA-FO approach. The latter was optimized in every experimental condition detailed above, and Fig. 2 presents the mean cluster metric of all parameters settings thus obtained. We report the optimal parameters setting found for each instance in Table 1.

We ran PA-CE with two mutation rates $m = \{0.01, 0.001\}$ in order to assess the importance of the single parameter m. Both PA-CE settings and PA-FO were run for 300 000 time-steps and over 100 independent runs.

The results in high density (i.e., Fig. 3) show that PA-CE can perform as well as – or better than – optimal parameters. The optimized controller scales up better than PA-CE with low mutations, but not perfectly. However compared to PA-CE, it requires a complete re-optimization of parameters for each experimental condition.

The critical parameter that determines performance in PA-CE appears to be the mutation rate. Indeed, PA-CE with higher mutation rate ($m = 0.01$) yields slightly lower efficiency in forming a single large cluster at low scales, but performs very well for large population sizes and even asymptotically outperforms the optimized controller for $N = 100$ (as confirmed by the significance of the Wilcoxon rank-sum test in the Supplementary Material), and $N = 200$ (which is clear visually, but is also corroborated statistically). Again, this superiority at large scales is obtained without the lengthy pre-tuning process that the optimized controller required.

These trends are even more visible at low density (see Table S2), which is inherently more challenging as agents therefore have a lower probability to encounter or form aggregates. This increased difficulty is manifested in the performance drop of PA-FO (see Fig. 4). Indeed, the latter visibly does not converge as fast as in high density conditions, and does not reach full aggregation with large population sizes, even though it was, each time, optimized for this specific setting. In comparison, PA-CE's performance is only slightly affected by the density decrease, especially in large populations, which makes the gap between PA-CE and PA-FO obvious for $N \geq 100$. Noticeably, the high mutation version, while it does not perform as well as the low mutation version in small populations, is the most scalable overall.

In fact, looking at Fig. 5, which shows the aggregation completion time of the swarm (where an aggregate is judged 'completed' when cluster metric ≥ 0.9), it appears that the high mutation version of PA-CE is superior both in terms of efficiency and consistency. Indeed, whatever the scale and density, it reaches full aggregation before any other model and achieves this with a consistent timing, as demonstrated by its comparatively low variability. In contrast, with $N \geq 50$, the low mutation version becomes less efficient and less consistent than even PA-FO.

In all the scenarios, the lack of scalability for low mutation rates is correlated with a lack of free agents (see Fig. 3 and 4) which, instead, settle in many small-to-mid-size aggregates—up to ten in large populations ($N = 200$). We hypothesize that the few mutations events happening are insufficient to disrupt even mid-size aggregates. Indeed, as the MNG favors words that are already known, the occasional mutation is quickly erased from lexicons by the overwhelming presence of a stable setting in the aggregate. In opposition, high mutation rates allow several simultaneous mutations which cause an increase in failed games and eventually disrupt even large aggregates, resulting in a larger proportion of free agents. This dynamics affects small populations negatively as even the biggest aggregate at that size is easily disrupted, but it is also the reason PA-CE can outperform a fixed, optimal aggregation controller. Indeed, in most probabilistic aggregation approaches, mid-size aggregates are often quite stable and difficult to break apart which means that large swarms tend to stabilize in several clusters, or take a very long time to leave that state. In those conditions, a higher mutation rate erodes these medium aggregates, accounting for the higher count of free agents, and allowing them to join another, bigger, aggregate in a shorter time.

4.2. Parameters selection

Heterogeneity is an important factor to PA-CE's success, most obviously by creating two ranges of behaviors: Firstly, a set of consistent settings, shared by entire aggregates and favoring the stability of that aggregate and, secondly, a diverse set (each free agent having an almost unique setting as they do not share it) favoring exploration. We therefore focused our attention to the emerging parameter values by re-processing the recordings of the experiments presented in Fig. 3. For each recorded time-steps (every one hundredth), we checked that 95% or more of the swarm had aggregated. If that was the case, we looked for a word that was in the lexicon of 95% or more agents and added that word to a list. This list was then ranked from the most to least occurring word.

In Fig. 6, we see that most of the selected words have values of b higher than 2 (whereas a and ρ are uniformly distributed, as seen also in Fig. S2 in the Supplementary Material, which

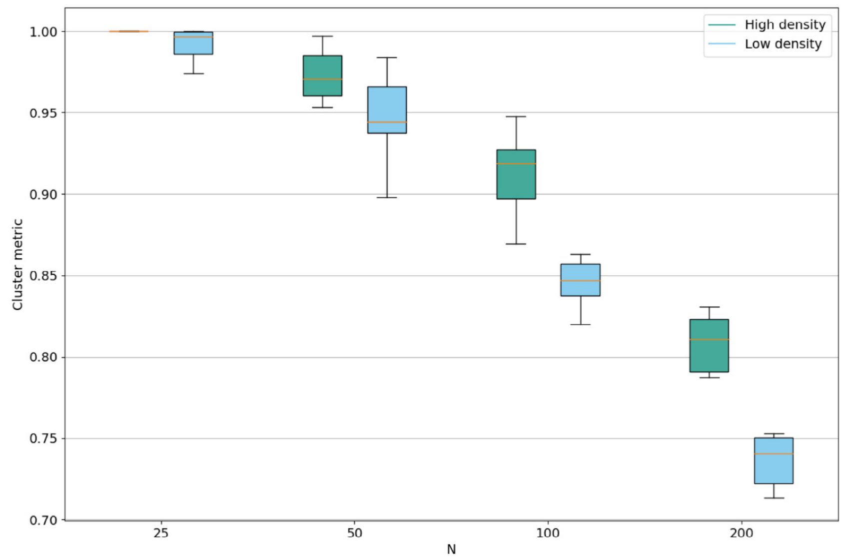


Fig. 2. Performance obtained over 10 optimization runs in irace for each configuration. A downward trend is clearly visible as the quantity of agents (and/or the density) increases (resp. decreases).

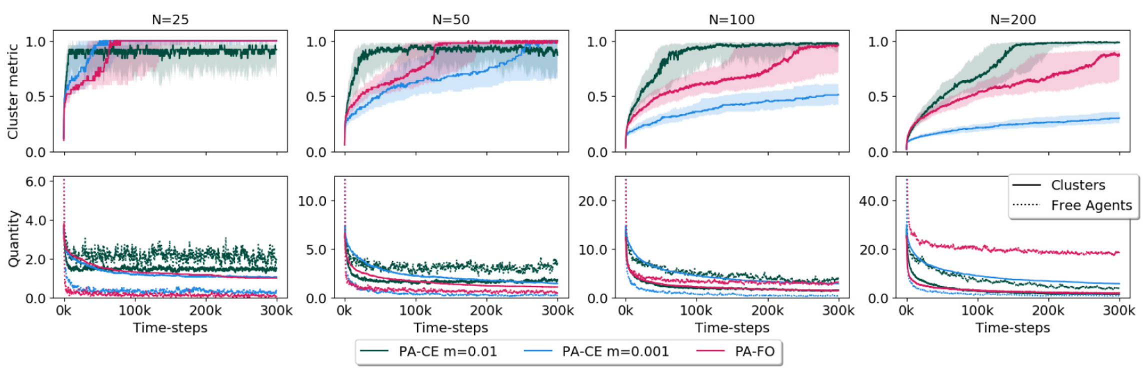


Fig. 3. Comparison between PA-CE and PA-FO in high density conditions, computed over 100 independent runs in populations of N agents. PA-CE is presented with two mutation settings $m = \{0.01, 0.001\}$. Experimental conditions were set to keep a constant density of 1 units per 12 square cell. Solid lines represent the median cluster metric and shaded regions indicate the 1st and 3rd quartiles.

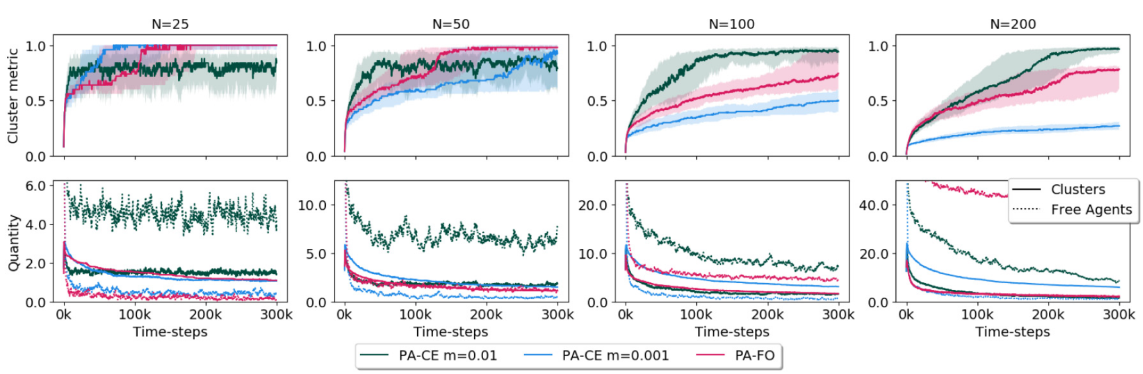


Fig. 4. Comparison between PA-CE and optimal fixed aggregation in low density conditions, computed over 100 independent runs in populations of N agents. PA-CE is presented with two mutation settings $m = \{0.01, 0.001\}$. Experimental conditions were set to keep a constant density of 1 units per 25 square cell. Solid lines represent the median cluster metric and shaded regions indicate the 1st and 3rd quartiles.

presents the distribution of parameter values). For reference, with $b = 2$, $p_{Leave}(1) \approx 0.135$ and $p_{Leave}(2) \approx 0.018$, meaning that the probability to leave an aggregate is very low already with 1 or 2 neighbors. Moreover, most words are evidently selected against as they only appear rarely and do not remain in the population, which is evident from the rapid decreasing of word occurrences (see second row of Fig. 6). This clearly showcases the behavior we expected: Parameters settings that encourage agents to leave aggregates are selected against, because the agents therefore leave and have a very low chance to infect the aggregate. Furthermore, the diversity of words starkly decreases as population size increases, which is explained by the topology of larger aggregates, which tends to filter out new words. Finally, in all population sizes, we observe an area around $b = 2.7$, where emerging

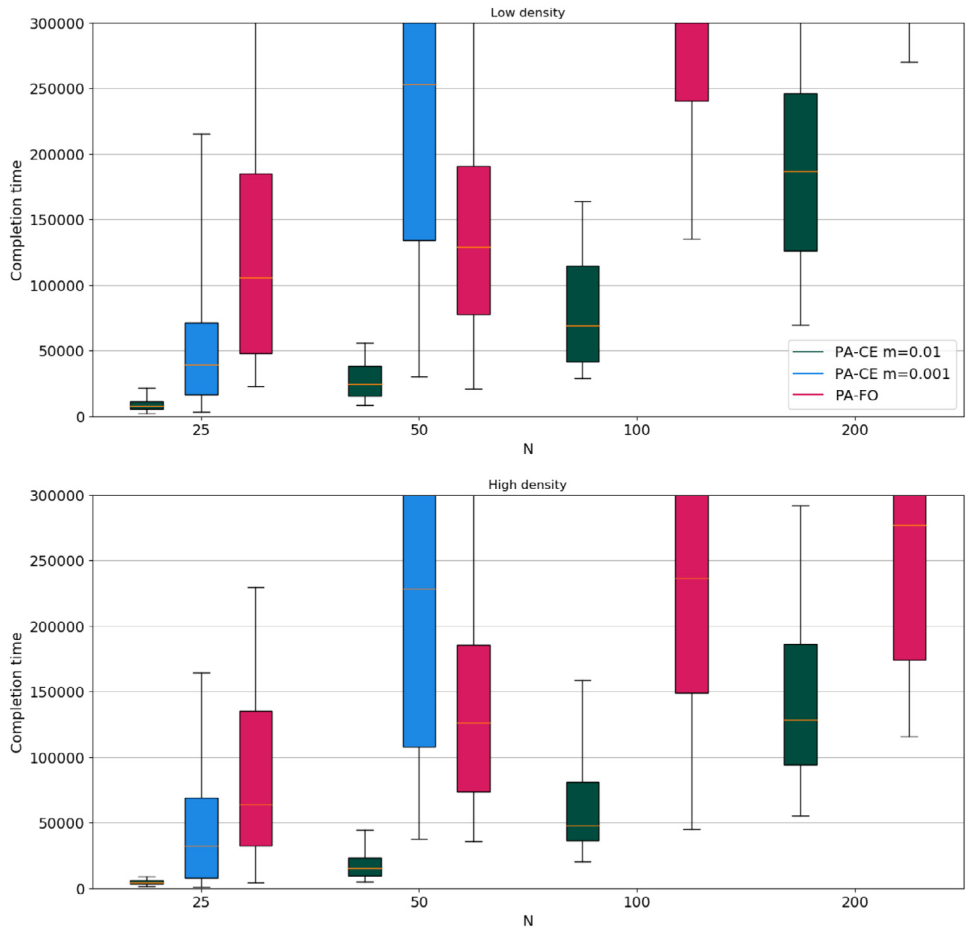


Fig. 5. Estimation of aggregation completion time in low (top) and high (bottom) swarm density, with swarm sizes $N = \{25, 50, 100, 200\}$, for two settings of PA-CE ($n = \{0.01, 0.001\}$) and PA-FO. As experiments were limited to a maximum time $t = 300000$, we used a Kaplan–Meier filter in order to include censored data. Low whisker, box bottom, yellow line, box top, and high whisker represent, resp., the 95th, 75th, 50th, 25th, and 5th of the survival curve, where survival means that cluster metric < 0.9. Elements not fitting into the frame all have a value of $+\infty$ as the survival curve converges above the corresponding percentile.

settings are much sparser. Interestingly, the most frequent settings in small populations seem to concentrate above this area, whereas the most frequent settings in large populations are found below. We can hypothesize that these are emergent categories of settings: one that seeks a trade-off between stability and exploration (below 2.7) and is favored by large aggregates (which, in classical aggregation, tend to be too stable), and another, that promotes stability, and is favored by small aggregates which are highly sensitive to mutations and, therefore, highly unstable by default. These two complementary properties – small aggregates are sensitive to mutations and unstable/large aggregates are stable and filter out mutations – are explained in Section 4.3.

4.3. Dispersion dynamics

The analysis above focuses on the selection of words that create stable aggregates. If only those were present in the system, it would just lead to the formation of stable but small aggregates. In order to explain the dispersion dynamics, we designed the following experimental protocol which highlights how formed clusters disband. We initialize agents as a tightly-packed aggregate in the middle of the arena. The algorithms were then run normally, leading some agents to leave the initial aggregate. Agents that reached the border of the arena were removed, in order to simulate capture by another aggregate or being lost in a very large area. Thus, the aggregate could only disperse and, eventually, disappear entirely. This experimental setting was used on PA-CE as well as on four settings of PA-FO (each optimized for a different size of $N = \{25, 50, 100, 200\}$). Each setting was run for 300 000 time-steps and over 100 independent runs per swarm size $N = \{25, 50, 75, 100, 125, 150, 175, 200\}$. As we can see in Fig. 7, different parameters settings yield similar dynamics for PA-FO as all the results, besides the one for PA-CE, are linear. This suggests that PA-FO performs similarly at all sizes (above 25) and fails to really implement the "larger aggregates are more stable" rule (i.e., they lose agents at the same rate). This is the consequence of the agents' limited neighborhood perception, which is already saturated for aggregates of size 25. Differently, PA-CE presents an increasingly monotonic non-linear curve, which means that larger aggregates are, indeed, increasingly and non-linearly more stable. Additionally, the variance observed with PA-CE is much larger than with PA-FO, meaning that even large aggregates can quickly disband with a non-negligible probability.

This non-linear increase in stability is the result of mutation avalanches that spread parameter settings to large portions of an aggregate. Fig. 8 presents results obtained with the same experimental protocol as before (i.e., starting as a packed aggregate and deleting agents that hit the border). However, here, only PA-CE was evaluated and each agent started with an identical parameters setting that provides strong cohesion. Moreover, mutations were only allowed for the first 500 time-steps. The experiment then played out, without mutations, until the aggregate had converged (i.e., all agents are back to sharing a single, identical, setting). As we can see in Fig. 8, larger aggregates are

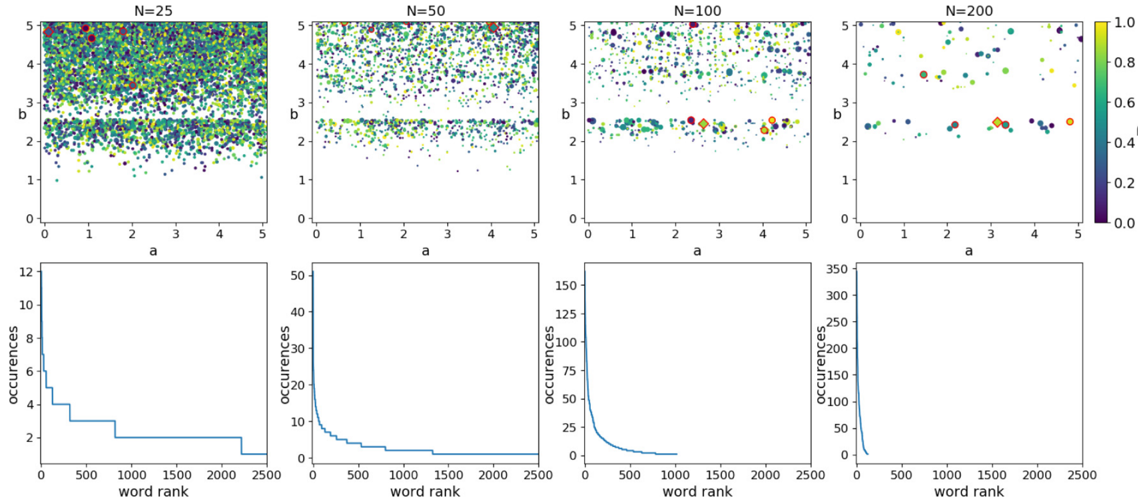


Fig. 6. The first row shows the parameter values of all the words that emerged in 100 runs of PA-CE with $N = 25, 50, 100, 200$ and a low population density. The size of the point show their normalized frequency. The five most frequent words are circled in red, the most frequent is diamond-shaped. The second row shows the total count of word apparition, ranked by descending order.

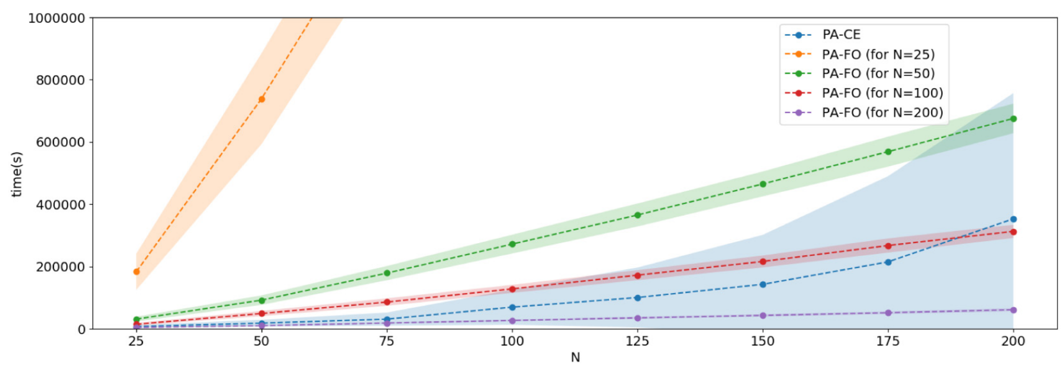


Fig. 7. Time (dot) and its standard deviation (shaded area) for all agents to leave the arena in a dispersion experiment, according to the aggregate size in 20 runs of the same experiments. Dotted dash lines to show trends.

non-linearly less sensitive to mutations. More accurately, the probability landscape has the shape of a switching function, i.e., it presents an abrupt change from high to low probability with respect to some threshold in the N dimension. This threshold itself, meanwhile, shifts right as the mutation rate increases. This is the result of a larger and more connected topology, which therefore supports consensus on a given setting much more effectively. Indeed, agents that are not directly in contact with a mutation can massively propagate the initial setting, pushing it back into the lexicon of the agents directly facing the mutations. Conversely, small aggregates are much more sensitive to mutations and can therefore easily adopt parameter settings that will break the aggregate. Such an event will be less probable in a large aggregate, as stability increases with respect to the size of aggregates.

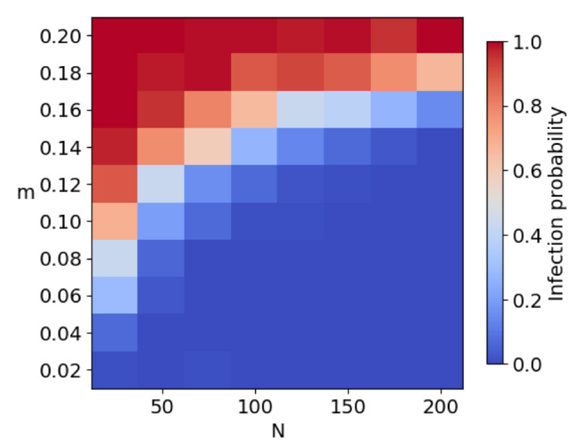


Fig. 8. Probability (computed from one hundred runs per setting) for a fully-converged aggregate of size N to convert to a new word when mutations at a rate m are introduced for 500 time-steps. The initial parameters setting shared by the whole population was $\{a = 2.74, b = 2.02, \rho = 0.9906\}$.

4.4. Embodied validation

As discussed in the previous sections, PA-CE with $m = 0.01$ is the most scalable setting. We thus used this parameter to validate our implementation on kilobots.

The ensuing performances, plotted in Fig. 9, continue to demonstrate a resilient behavior at all scales, which is a notable challenge in self-organized aggregation, especially as the population density increases proportionally [19]. As in the simulated experiments, PA-CE maintains a significant quantity of free agents, which prevents small populations from forming a proportionally large aggregate. This is obvious in Fig. 10, which illustrates an experiment with 25 kilobots and shows that they form aggregates fast, which experience regular disbanding events afterwards, eventually leaving only one stable aggregate. The higher the density, the faster the aggregation and the larger and

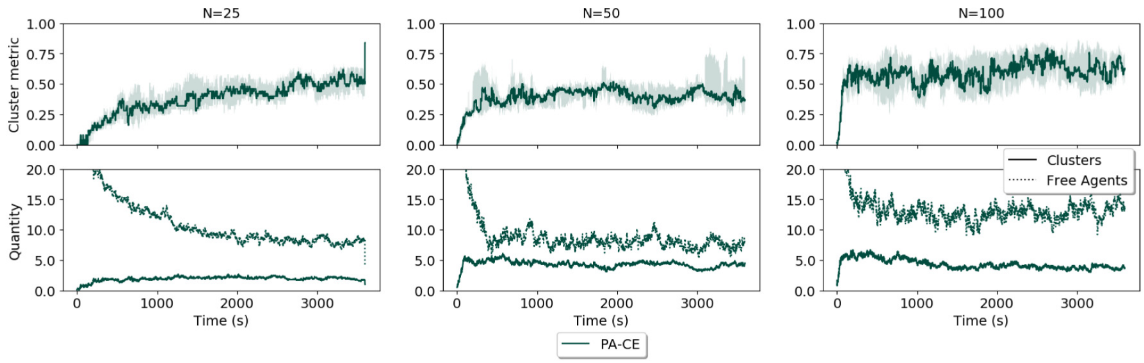


Fig. 9. Performances of PA-CE implemented on swarms of N kilobots.

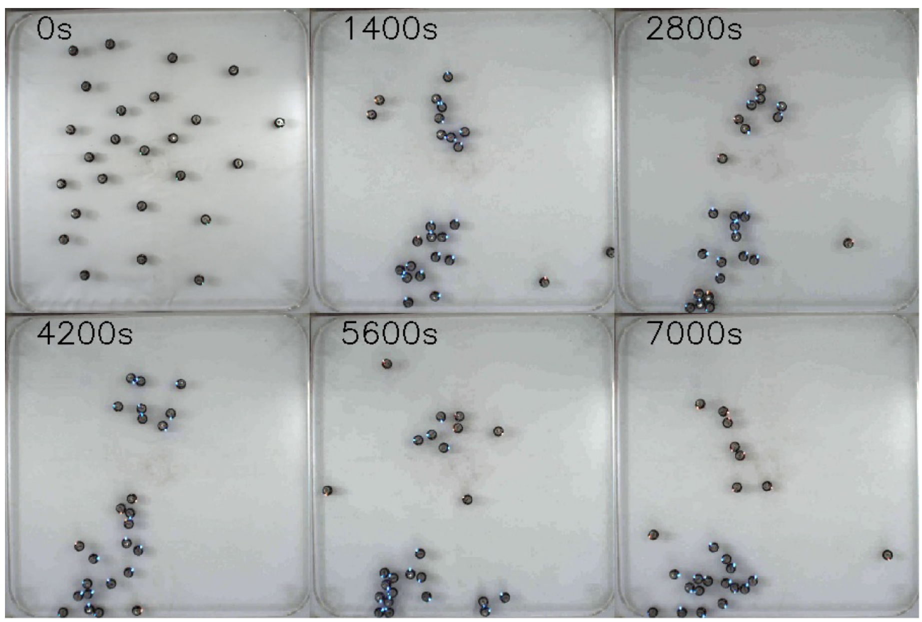


Fig. 10. Steps of a PA-CE experiment with 25 kilobots, as watchable in the Supplementary Material Movie S2.

more stable the aggregates. This confirms the results from simulations, although a reduction in the cluster metrics is observed, which can be justified by the noisy communication experienced by real kilobots, that makes the measure of the number of neighbors somewhat unreliable, missing to recognize some of them in several cases.

5. Conclusions

Experimental results showed that PA-CE alters the dynamics of aggregation by introducing dynamic and heterogeneous parameters, which continuously vary due to mutations across the swarm. Differently from other swarm robotic behaviors, whereby rules are fixed but can create different dynamics depending on local interactions and environmental conditions, here, the rules themselves change according to these factors, which reinforces the plasticity of the swarm's behavior.

In the case of self-organized aggregation, this yields more scalable dynamics, first by enabling two classes of behaviors: one supporting exploration, and the other favoring aggregate stability and growth. Secondly, even within an aggregate, the rules can still change depending on macroscopic factors that are outside the sensing range of individual agents. Indeed, in normal aggregation, the decision to leave an aggregate only depends on local information and, therefore, does not change as long as the aggregate size is larger than the agent's sensing capabilities. In comparison, in PA-CE macroscopic features such as the entire size and shape of the aggregate determines the ability of mutations to spread widely and to determine change in the behavior of individuals.

Consequently, with respect to self-organized aggregation, we have introduced a new model which requires very limited abilities, namely (i) perceive and count other agents within some range, and (ii) exchange a few bytes of information. Our model outperforms existing algorithms with similar limitations in terms of speed of aggregation and aggregate size. Furthermore, this approach could be extended to other tasks in order to similarly improve existing models without increasing deployment time (differently from offline optimization approaches). Indeed, the dynamics we observed in PA-CE displayed a remarkably adaptive handling of the exploitation–exploration (in this case, staying–leaving) trade-off. We suggest that PA-CE could be extended in order to create a design pattern, X-CE, for any behavior that could be modeled as a PFSM with a state to explore and a state to exploit. For example, we could address the problem of area coverage [67–69] with an algorithm very similar to PA-CE, with the addition that communication noise increases as the distance between robots reduces. Proximity between robots would therefore favor the emergence of new behaviors, whereas well-distributed swarm would be more stable. This project is facilitated by the design of our cultural evolution framework, which only requires limited communications and relies entirely on their situatedness [70]. In that sense it is perfectly adapted to most swarm robotic platforms, as demonstrated by our implementation on the very minimal kilobots [23].

This is a novel approach to the design problem in swarm robotics. It is a distributed and embodied method, contrary to offline automatic design [54,55,71,72], brute-force search [21], or adaptive response threshold approaches [12]. PA-CE does not, indeed, require any offline tuning (though it involves a single mutation parameter that can be set very broadly). Similarly to embodied evolution approaches [11], it encourages behavioral diversity. Our results show that this diversity, coupled with our unique macro-to-micro feedback leads to more resilient behaviors than homogeneous optimized controllers, which suggests that optimization alone is not sufficient for an efficient design method. A more detailed analysis of these unique dynamics could be obtained through mathematical models for swarm aggregation [50,73,74] coupled with social dynamics [24,41,45]. Such an analysis should provide important answers with regards to the roles of diversity and dynamic parameters settings in distributed behaviors efficiency, compared to static behaviors.

This work is part of a larger research goal to create more flexible and adaptive communication in swarm robotics in order to create truly autonomous swarm systems [42]. This new approach builds on Language Games (LGs) to create language-like communication, which would, eventually, give swarms rich vocabularies to define their environment [44,46], as well as their task, dynamically. PA-CE, by its use of a LG, is inscribed in this proposition and can be conceived of as a first step to verbalizing actions and creating 'language interwoven with an activity' [75], whereas existing implementation of LG still focus on very 'linguistic' tasks such as naming or categorizing [76].

CRediT authorship contribution statement

Nicolas Cambier: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft and editing, Visualization. **Dario Albani:** Software, Validation, Investigation. **Vincent Frémont:** Supervision, Writing – review & editing. **Vito Trianni:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Eliseo Ferrante:** Conceptualization, Methodology, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2021.108010.

References

[1] A. Khuong, J. Gautrais, A. Perna, C. Sbaï, M. Combe, P. Kuntz, C. Jost, G. Theraulaz, Stigmergic construction and topochemical information shape ant nest architecture, Proc. Natl. Acad. Sci. 113 (5) (2016) 1303–1308, http://dx.doi.org/10.1073/pnas.1509829113.

[2] K. Singh, B.P. Muljadi, A.Q. Raeini, C. Jost, V. Vandeginste, M.J. Blunt, G. Theraulaz, P. Degond, The architectural design of smart ventilation and drainage systems in termite nests, Sci. Adv. 5 (3) (2019) http://dx.doi.org/10.1126/sciadv.aat8520.

[3] T.D. Seeley, P.K. Visscher, T. Schlegel, P.M. Hogan, N.R. Franks, J.A.R. Marshall, Stop signals provide cross inhibition in collective decision-making by honeybee swarms, Science 335 (6064) (2012) 108–111, http://dx.doi.org/10.1126/science.1210361, URL https://science.sciencemag.org/content/335/6064/108.

[4] A. Reina, T. Bose, V. Trianni, J.A.R. Marshall, Psychophysical laws and the superorganism, Sci. Rep. 8 (1) (2018) 4387–4388, http://dx.doi.org/10.1038/s41598-018-22616-y.

[5] S. Camazine, Self-Organization in Biological Systems, Princeton University Press, 2003.

[6] C. Detrain, J.-L. Deneubourg, Self-organized structures in a superorganism: Do ants "behave" like molecules? Physics of Life Reviews 3 (3) (2006) 162–187.

[7] M. Dorigo, G. Theraulaz, V. Trianni, Reflections on the future of swarm robotics, Science Robotics 5 (49) (2020) eabe4385, http://dx.doi.org/10.1126/scirobotics.abe4385.

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

[9] M. Rubenstein, A. Cornejo, R. Nagpal, Programmable self-assembly in a thousand-robot swarm, Science 345 (6198) (2014) 795–799.

[10] M. Dorigo, D. Floreano, L.M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, et al., Swarmanoid: A novel concept for the study of heterogeneous robotic swarms, IEEE Robot. Autom. Mag. 20 (4) (2013) 60–71.

[11] I.F. Pérez, A. Boumaza, F. Charpillet, Maintaining diversity in robot swarms with distributed embodied evolution, in: International Conference on Swarm Intelligence, Springer, 2018, pp. 395–402.

[12] E. Castello, T. Yamamoto, F. Dalla Libera, W. Liu, A.F. Winfield, Y. Nakamura, H. Ishiguro, Adaptive foraging for simulated and real robotic swarms: The dynamical response threshold approach, Swarm Intell. 10 (1) (2016) 1–31.

[13] G.A. van Kleef, M.J. Gelfand, J. Jetten, The dynamic nature of social norms: New perspectives on norm development, impact, violation, and enforcement, J. Exp. Soc. Psychol. 84 (2019) 103814, http://dx.doi.org/10.1016/j.jesp.2019.05.002.

[14] D. Centola, A. Baronchelli, The spontaneous emergence of conventions: An experimental study of cultural evolution, Proc. Natl. Acad. Sci. 112 (7) (2015) 1989–1994, http://dx.doi.org/10.1073/pnas.1418838112.

[15] D. Centola, J. Becker, D. Brackbill, A. Baronchelli, Experimental evidence for tipping points in social convention, Science 360 (6393) (2018) 1116–1119, http://dx.doi.org/10.1126/science.aas8827.

[16] B. Morsky, E. Akçay, Evolution of social norms and correlated equilibria, Proc. Natl. Acad. Sci. 116 (18) (2019) 8834–8839, http://dx.doi.org/10.1073/pnas.1817095116.

[17] L. Steels, A self-organizing spatial vocabulary, Artif. Life 2 (3) (1995) 319–332.

[18] O. Soysal, E. Sahin, Probabilistic aggregation strategies in swarm robotic systems, in: Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, IEEE, 2005, pp. 325–332.

[19] L. Bayindir, E. Sahin, Modeling self-organized aggregation in swarm robotic systems, in: Swarm Intelligence Symposium, 2009. SIS'09. IEEE, IEEE, 2009, pp. 88–95.

[20] V. Trianni, A. Campo, Fundamental collective behaviors in swarm robotics, in: Springer Handbook of Computational Intelligence, Springer, 2015, pp. 1377–1394.

[21] M. Gauci, J. Chen, W. Li, T.J. Dodd, R. Groß, Self-organized aggregation without computation, Int. J. Robot. Res. 33 (8) (2014) 1145–1161.

[22] B. Khaldi, F. Harrou, F. Cherif, Y. Sun, Flexible and efficient topological approaches for a reliable robots swarm aggregation, IEEE Access 7 (2019) 96372–96383.

[23] M. Rubenstein, C. Ahler, R. Nagpal, Kilobot: A low cost scalable robot system for collective behaviors, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, 2012, pp. 3293–3298.

[24] C. Castellano, S. Fortunato, V. Loreto, Statistical physics of social dynamics, Rev. Modern Phys. 81 (2) (2009) 591–646.

[25] G. Valentini, E. Ferrante, M. Dorigo, The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives, Front. Robot. AI 4 (2017) 9.

[26] C.A. Parker, H. Zhang, Cooperative decision-making in decentralized multiple-robot systems: The best-of-n problem, IEEE/ASME Trans. Mechatronics 14 (2) (2009) 240–251.

[27] G. Valentini, E. Ferrante, H. Hamann, M. Dorigo, Collective decision with 100 kilobots: Speed versus accuracy in binary discrimination problems, Auton. Agents Multi-Agent Syst. (2015) 1–28, URL http://dx.doi.org/10.1007/s10458-015-9323-3.

[28] A. Scheidler, A. Brutschy, E. Ferrante, M. Dorigo, The k-unanimity rule for self-organized decision-making in swarms of robots, IEEE Trans. Cybern. 46 (5) (2015) 1175–1188.

[29] M.A.M. De Oca, E. Ferrante, N. Mathews, M. Birattari, M. Dorigo, Opinion dynamics for decentralized decision-making in a robot swarm, in: International Conference on Swarm Intelligence, Springer, 2010, pp. 251–262.

[30] G. Valentini, H. Hamann, M. Dorigo, et al., Self-organized collective decision making: The weighted voter model, in: AAMAS, 2014, pp. 45–52.

[31] J.A.R. Marshall, R. Bogacz, A. Dornhaus, R. Planqué, T. Kovacs, N.R. Franks, On optimal decision-making in brains and social insect colonies, J. Roy. Soc. Interface Roy. Soc. 6 (40) (2009) 1065–1074, http://dx.doi.org/10.1098/rsif.2008.0511.
[32] M.A.M. de Oca, E. Ferrante, A. Scheidler, C. Pinciroli, M. Birattari, M. Dorigo, Majority-rule opinion dynamics with differential latency: A mechanism for self-organized collective decision-making, Swarm Intell. 5 (3–4) (2011) 305–327.
[33] A. Reina, R. Miletitch, M. Dorigo, V. Trianni, A quantitative micro–macro link for collective decisions: The shortest path discovery/selection example, Swarm Intell. 9 (2–3) (2015) 75–102.
[34] D. Albani, T. Manoni, D. Nardi, V. Trianni, Dynamic UAV swarm deployment for non-uniform coverage, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 523–531.
[35] L. Steels, Modeling the cultural evolution of language, Phys. Life Rev. 8 (4) (2011) 339–356.
[36] L. Steels, Modeling the formation of language in embodied agents: Methods and open challenges, in: Evolution of Communication and Language in Embodied Agents, Springer, 2010, pp. 223–233.
[37] A. Baronchelli, M. Felici, V. Loreto, E. Caglioti, L. Steels, Sharp transition towards shared vocabularies in multi-agent systems, J. Stat. Mech. Theory Exp. 2006 (06) (2006) P06014.
[38] L. Steels, M. Loetzsch, The grounded naming game, in: Experiments in Cultural Language Evolution, John Benjamins, 2012, pp. 41–59.
[39] A. Baronchelli, The emergence of consensus: A primer, Roy. Soc. Open Sci. 5 (2) (2018) 172189.
[40] V. Loreto, A. Baronchelli, A. Puglisi, Mathematical modeling of language games, in: Evolution of Communication and Language in Embodied Agents, Springer, 2010, pp. 263–281.
[41] A. Baronchelli, Role of feedback and broadcasting in the naming game, Phys. Rev. E 83 (4) (2011) 046103.
[42] N. Cambier, R. Miletitch, V. Frémont, M. Dorigo, E. Ferrante, V. Trianni, Language evolution in swarm robotics: A perspective, Front. Robot. AI 7 (2020) 12.
[43] V. Trianni, D. De Simone, A. Reina, A. Baronchelli, Emergence of consensus in a multi-robot network: From abstract models to empirical validation, IEEE Robot. Autom. Lett. 1 (1) (2016) 348–353.
[44] R. Miletitch, A. Reina, M. Dorigo, V. Trianni, Emergent naming of resources in a foraging robot swarm, 2019, arXiv preprint arXiv:1910.02274.
[45] V. Loreto, A. Baronchelli, A. Mukherjee, A. Puglisi, F. Tria, Statistical physics of language dynamics, J. Stat. Mech. Theory Exp. P04006 (2011) http://dx.doi.org/10.1088/1742-5468/2011/04/P04006.
[46] N. Cambier, V. Frémont, E. Ferrante, Group-size regulation in self-organised aggregation through the naming game, in: International Symposium on Swarm Behavior and Bio-Inspired Robotics, SWARM 2017, Kyoto, Japan, 2017, URL https://hal.archives-ouvertes.fr/hal-01679600.
[47] N. Cambier, V. Frémont, V. Trianni, E. Ferrante, Embodied Evolution of Self-Organised Aggregation by Cultural Propagation in: ANTS 2018, Rome, Italy, 2018.
[48] S. Garnier, C. Jost, R. Jeanson, J. Gautrais, M. Asadpour, G. Caprari, G. Theraulaz, Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots, in: European Conference on Artificial Life, Springer, 2005, pp. 169–178.
[49] A. Campo, S. Garnier, O. Dédriche, M. Zekkri, M. Dorigo, Self-organized discrimination of resources, PLoS One 6 (5) (2011) e19888.
[50] N. Correll, A. Martinoli, Modeling and designing self-organized aggregation in a swarm of miniature robots, Int. J. Robot. Res. 30 (5) (2011) 615–626.
[51] M. Bodi, R. Thenius, M. Szopek, T. Schmickl, K. Crailsheim, Interaction of robot swarms using the honeybee-inspired control algorithm BEECLUST, Math. Comput. Model. Dyn. Syst. 18 (1) (2012) 87–100.
[52] C. Dimidov, G. Oriolo, V. Trianni, Random walks in swarm robotics: An experiment with kilobots, in: M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, T. Stützle (Eds.), Swarm Intelligence: 10th International Conference, ANTS 2016, Brussels, Belgium, September 7-9, 2016, Proceedings, Springer International Publishing, 2016, pp. 185–196.
[53] Y. Katada, Evolutionary design method of probabilistic finite state machine for swarm robots aggregation, Artif. Life Robot. 23 (4) (2018) 600–608.
[54] V. Trianni, R. Groß, T.H. Labella, E. Şahin, M. Dorigo, Evolving aggregation behaviors in a swarm of robots, in: European Conference on Artificial Life, Springer, 2003, pp. 865–874.
[55] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T.H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, et al., Evolving self-organizing behaviors for a swarm-bot, Auton. Robots 17 (2) (2004) 223–245.
[56] E. Şahin, Swarm robotics: From sources of inspiration to domains of application, in: International Workshop on Swarm Robotics, Springer, 2004, pp. 10–20.
[57] A.L. Nelson, G.J. Barlow, L. Doitsidis, Fitness functions in evolutionary robotics: A survey and analysis, Robot. Auton. Syst. 57 (4) (2009) 345–370.
[58] M. Coppola, J. Guo, E. Gill, G. de Croon, The PageRank algorithm as a method to optimize swarm behavior through local analysis, Swarm Intell. (2019) 1–43.
[59] N. Bredeche, E. Haasdijk, A. Prieto, Embodied evolution in collective robotics: A review, Front. Robot. AI 5 (2018) 12.
[60] F. Silva, P. Urbano, L. Correia, A.L. Christensen, odNEAT: An algorithm for decentralised online evolution of robotic controllers, Evolutionary Computation 23 (3) (2015) 421–449.
[61] A. Leccese, A. Gasparri, A. Priolo, G. Oriolo, G. Ulivi, A swarm aggregation algorithm based on local interaction with actuator saturations and integrated obstacle avoidance, in: 2013 IEEE International Conference on Robotics and Automation, IEEE, 2013, pp. 1865–1870.
[62] C. Shannon, Mathematical theory of communication, Bell Syst. Tech. J. (1948).
[63] N. Bredeche, J.-M. Montanier, Environment-driven embodied evolution in a population of autonomous agents, in: International Conference on Parallel Problem Solving from Nature, Springer, 2010, pp. 290–299.
[64] N. Noskov, E. Haasdijk, B. Weel, A.E. Eiben, MONEE: Using parental investment to combine open-ended and task-driven evolution, in: European Conference on the Applications of Evolutionary Computation, Springer, 2013, pp. 569–578.
[65] K. Hasselmann, A. Ligot, J. Ruddick, M. Birattari, Empirical assessment and comparison of neuro-evolutionary methods for the automatic off-line design of robot swarms, Nature Commun. 12 (1) (2021) 1–11.
[66] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, T. Stützle, M. Birattari, The irace package: Iterated racing for automatic algorithm configuration, Oper. Res. Perspect. 3 (2016) 43–58, http://dx.doi.org/10.1016/j.orp.2016.09.002.
[67] A. Howard, M.J. Matarić, G.S. Sukhatme, Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem, in: Distributed Autonomous Robotic Systems 5, Springer, 2002, pp. 299–308.
[68] S. Rutishauser, N. Correll, A. Martinoli, Collaborative coverage using a swarm of networked miniature robots, Robot. Auton. Syst. 57 (5) (2009) 517–525.
[69] E. Ventocilla, Swarm-based area exploration and coverage based on pheromones and bird flocks, 2013.
[70] A. Gutiérrez, A. Campo, M. Dorigo, D. Amor, L. Magdalena, F. Monasterio-Huelin, An open localization and local communication embodied sensor, Sensors 8 (11) (2008) 7545–7563.
[71] G. Francesca, M. Brambilla, A. Brutschy, L. Garattoni, R. Miletitch, G. Podevijn, A. Reina, T. Soleymani, M. Salvaro, C. Pinciroli, et al., Automode-Chocolate: Automatic design of control software for robot swarms, Swarm Intell. 9 (2–3) (2015) 125–152.
[72] G. Francesca, M. Birattari, Automatic design of robot swarms: Achievements and challenges, Front. Robot. AI 3 (2016) 29.
[73] J.-M. Amé, C. Rivault, J.-L. Deneubourg, Cockroach aggregation based on strain odour recognition, Anim. Behav. 68 (4) (2004) 793–801.
[74] Z. Firat, E. Ferrante, Y. Gillet, E. Tuci, On self-organised aggregation dynamics in swarms of robots with informed robots, 2019, arXiv preprint arXiv:1903.03841.
[75] L. Wittgenstein, Philosophical investigations, trans, GEM Anscombe, 261, 1953, p. 49.
[76] L. Steels, Language games for autonomous robots, IEEE Intell. Syst. 16 (5) (2001) 16–22.