Group-size Regulation in Self-Organised Aggregation through the Naming Game

Nicolas Cambier[1†], Vincent Frémont[1] and Eliseo Ferrante[2]

[1]Sorbonne universités, Université de technologie de Compiègne,
UMR CNRS 7253 Heudiasyc, Compiègne, France
(E-mail: firstname.lastname@hds.utc.fr)
[2]Laboratory of Socio-ecology and Social Evolution, KU Leuven, Leuven, Belgium
(E-mail: eliseo.ferrante@kuleuven.be)

Abstract:
Language games are self-organised models of the evolution of language used in the context of natural language evolution. They are used to study the emergence of a shared vocabulary through a self-organisation process. These models have been traditionally tested in settings where the interaction topology among individuals is mostly static. Only recently, these models have been introduced to the context of swarm robotics, in order to study the effect of embodiment and random agent mobility on the evolution of language. These results have shown that, even in such setting, the naming game exhibits the same outcome as in simpler simulations: all agents achieve consensus on a single word.
In this paper, we study the interaction effect between the naming game and one of the simplest, yet most important collective behaviour studied in swarm robotics: self-organised aggregation. This collective behaviour can be seen as the building blocks for many others, as it is required in order to gather robots, unable to sense their global position, at a single location. Achieving this collective behaviour is particularly challenging, especially in environments without landmarks. Here, we augment a classical aggregation algorithm with a naming game model. Experiments reveal that this combination extends the capabilities of the naming game as well as of aggregation: It allows the emergence of more than one word, and allows aggregation to form a controllable number of groups. These results are very promising in the context of collective exploration, as it allows robots to divide the environment in different portions and at the same time give a name to each portion, which can be used for more advanced subsequent collective behaviours.

Keywords: Self-organised aggregation, group-size regulation, language game, naming game, swarm robotics

1. INTRODUCTION

Swarm robotics takes inspiration from studies in social animals and aims at designing collective behaviours for large groups of robots, that have to cooperate in order to solve a task by only relying on local sensing and communication [1]. A central aspect in swarm robotics is (local) communication, that is one of the main coordination mechanisms used to achieve a collective behaviour. Apart from approaches that include communication implicitly as a component of the individual behaviour leading to the desired collective behaviour, other studies have more explicitly focused on the emergence of communication itself within a swarm, focusing on questions such as: ”When is communication needed in the first place?”, ”How does language emerge?”, and ”Which form should communication take in order to be most effective?”. The main motivation for studying these questions in swarm robotics is to allow robot swarms to tackle novel situations in ways that may not be *a priori* obvious to the experimenter. This is a necessary feature for having swarms that are fully autonomous.

Most of the work done in the above direction belongs to the framework of evolutionary swarm robotics [2, 3], which studies the application of biological evolutionary models to swarm robotics. One of the first attempts in this direction observed the emergence of simple communication through the evolution of individual behaviours within small colonies of robots in a foraging scenario [4]. The environment hosted undistinguishable food and poison sources which both emitted red light. By evolving a neural network, the colonies developed simple communication by activating blue lights either near the food sources or near the poison sources in order to signal it to their comrades. Not only did these communication schemes emerge, but they actually improved the efficiency of the colonies compared to colonies that were not equipped with lights. These results demonstrated that communication can provide an advantage to robots swarms in a typical scenario of swarm robotics.

Other models of emerging communication exist, which are traditionally studied outside of swarm robotics to study the evolution of languages in humans [5]. One of such models is the Language Game [6], which was originally devised to study the formation of a shared lexicon as the result of self-organisation [7]. In its simplest instance, the Naming Game [8], two agents use a simple communication protocol in order to name (i.e. decide of a word for) a predetermined topic. The Naming Game shows how self-organisation can result in a *shared* and *efficient communication system* **without** *any generational transmission* [9]. A later variation, the Minimal Naming Game [10], allows simultaneous games in large populations using broadcasting instead of one-to-one communication. Using the minimal naming game within swarm robotics

† Nicolas Cambier is the presenter of this paper.

is a meaningful research direction, as it can be employed by a swarm to label different choices with a word, and this can later be used in a collective decision-making scenario, in order for the swarm to choose the best among these options [11, 12].

The Minimal Naming Game has recently been introduced into the field of Swarm Robotics by Trianni et al. [13] who implemented it in a swarm of kilobots [14]. The swarm was not engaged in any collective behaviour. Rather, the robots were all executing an individual random walk. The main aim of this study was to investigate whether the dynamics of the naming game change when implemented on agents that are mobile and embodied. The major conclusion from this work is that the embodiment of agents playing a naming game reduces the strain on their memory as the collision between transmissions results in a loss of data (and thus the abortion of a part of the games). Conversely these collisions lead to the formation of aggregates of robots that do not interact much, leading to slower convergence than with simulated agents. Despite this, the algorithm still makes the swarm converge to a single word.

In this paper, we also aim at studying the effect of the Minimal Naming Game in a swarm robotics setting. Differently from [13], the aim here is to study what happens when the swarm is undergoing the dynamics dictated by a collective, coordinated behaviour, rather than engaged in an individual random walk. In this regard, we used self-organised aggregation as an initial case study as it is perhaps the simplest collective behaviour to study, and at the same time it can be seen as a prerequisite for other forms of cooperation and, thus, for other tasks in Swarm Robotics [15].

Self-organised aggregation is a decision-making process whereby robots need to gather all around the *same* area, without relying to global information, global communication, or any kind of centralised information or decision. Early studies on aggregation were inspired from cockroach larvae's behaviour [16]. In this model, individuals (be they cockroaches or robots) have to collectively choose an area as shelter, where a shelter is an area in the environment that can be clearly perceived by all agents. In order to choose a shelter, each individual explores the entire environment randomly until it finds a shelter and stops. By only using this mechanism, individuals would aggregate randomly in all of the shelters. Thus, a second mechanism is required to reach a collective decision on which area to select as the shelter. This mechanism is based on a probabilistic stopping and leaving criteria, with probabilities modelled after real cockroaches and reported in [17]. This model has inspired many swarm robotics algorithms. The closest study was those of Garnier et al. [18], which studied a very similar aggregation task with shelters, and even used the very same probability table as in [17]. Other works have studied self-organised aggregation without requiring the presence of shelters in the environment [19-21] according to the observations in [17]. In all these cases, the behaviour of these robots was modelled by a probabilistic finite state automaton (PFSA) which decides whether to leave or join an aggregate. Abandoning the biologically-inspired approaches of evolutionary robotics or ethology, a later model [22] of aggregation reached impressive results with deterministically controlled robots equipped with a single one-bit sensor, without using any PFSA but only a simple mapping sensors-actuators.

In this paper, we study the mutual interaction between the minimal naming game and self-organised aggregation as inspired by the cockroach collective behaviour. The details of our model are presented in Section 2. We then detail our experimental setup in Section 3 and analyse the dynamics of this new algorithm in Section 4. Our conclusions and propositions of future work are exposed in Section 5.

2. MODEL

In this section, we explain the details of our naming-game augmented aggregation model. In Section 2.1, we first present the basic aggregation model, without the naming game. We then describe how the naming game has been introduced within aggregation.

2.1. Aggregation Mechanism

Our self-organised aggregation approach is also modelled after the behaviour of cockroaches' larvae [17]. This mechanism only requires robots that are capable of estimating the number of neighbours in range of sight.

The model works as follows; Individuals can either explore the environment or stay stationary. This is implemented in a PFSA whereby individuals can be in one of two states, $WALK$ and $STAY$. The probability of remaining in the $STAY$ state is proportional to the estimated aggregate size around the robot.

The description up to this point makes our algorithm essentially identical to the one proposed in [18]. However, in our experiments we want to study self-organised aggregation without the need of environmental landmarks such as shelters. As the model in [18] required shelters, we had to introduce some modifications in order to have it working without shelters. To this end, our algorithm, whose PFSA is presented in Fig. 1, includes the following three modifications:

- In the original approach [18], the probability to stay in an aggregate was determined by a table published in [17]. Here, we use a linear probability function (αP_{Base}, where P_{Base} is a configurable probability). This strengthens the emerging of aggregates as this probability reaches 1 eventually with large neighbourhood sizes, whereas in the original table this probability saturated to a value $\ll 1$.
- Explicitly configurable number of time-steps between two consecutive sampling of probabilities. When in the state $STAY$, the probability to leave is sampled once every T_{STAY} time-steps.
- Introduction of a $LEAVE$ state before resuming nor-

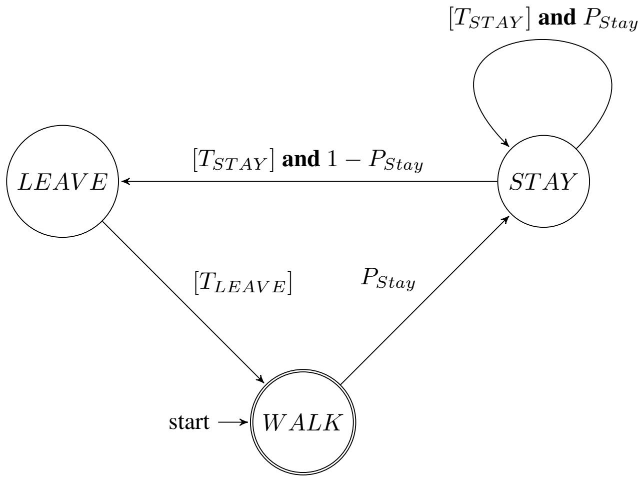

Fig. 1 PFSA of our aggregation mechanism. A parametrisable number of time-steps elapses between any state transition. P_{Stay} is calculated as a base probability (P_{Base}) multiplied by the number of neighbours in the $STAY$ state. Values inside square brackets $[T]$ indicate that T time-steps have passed since the PFSA entered the state.

mal random walking, to make sure the robots go far enough from their previous shelter (i.e. they do not go back to the $STAY$ state too soon), thus improving exploration. The robots stay in the $LEAVE$ state exactly T_{LEAVE} time-steps.

2.2. Aggregation augmented with the Minimal Naming Game

The original Naming Game [7] is played by two agents, chosen randomly in the population, who must agree on a word to name a predetermined topic. In order to reach this goal, the agents, who both possess an individual lexicon, will take different roles; one will be the **speaker** and the other one will be the **hearer**. The speaker utters a random word from its lexicon (or, if the lexicon is empty, a randomly generated word). If the hearer already knows this word, it increases its association score with this word. Else, it adds the words to its lexicon. In any case, the hearer notifies the speaker of whether it already knew the word or not. In the first case, the game is deemed successful and the speaker increases the word's association score too. In the second case, the game fails and the speaker decreases the association score.

A minimal variations of this game is possible [23] as, in case of success, the speaker and hearer can both drop all words associated to the topic except the one that just provoked the success. Furthermore, mathematical analyses [10] of the spread of the words in a naming game with different update scheme (i.e. when both agents update their lexicon after the game or when only the hearer/speaker update its lexicon) demonstrated that the speaker do not actually need to update its lexicon after the game. As a consequence, it appears that the hearer does not need to communicate the success or failure of the game which, in turns, renders this step useless. Eventually, it means that the naming game can be played with several hearers at once as the speaker can just broadcast its chosen word without needing to wait for any answer. Thus, a minimal naming game can proceed as described in Algo. 1 (where $Lexicon$ is a list of words heard by the agent). The game is deemed successful if the speaker broadcasts a word already in the hearer's lexicon.

From there, the Minimal Naming Game can straightforwardly be coupled with our aggregation algorithm by following three rules:

- Robots in the $STAY$ state play the role of the **speaker**.
- Robots that can transition from $WALK$ to $STAY$ or from $STAY$ to $LEAVE$ play the **hearers**.
- The probability function is now dependent on the number of neighbours in the $STAY$ state (as before), but also on the number of successful naming game with them in a row. For each neighbour, if the naming game is successful, the count is incremented. However, if the naming game is not successful, the count is reset as the lexicon is no more limited to a single word. This ensures that the neighbours counted are all uttering the same word.

The PFSA remains identical even though the behaviour of the robots are slightly more complex. The resulting algorithm is explained in Algo. 2.

It should be underlined that, if the whole swarm eventually agrees on a single word (as it normally does [9]), then this algorithm would eventually behave exactly as the game-less version presented in Section 2.1. Indeed, this single word would then act as a simple signal signifying that the robot is in the $STAY$ state.

3. EXPERIMENTAL SETUP

The new algorithm described in Section 2 can help us answer two questions: "How does the Naming Game influence aggregation?" and "How does aggregation influence the Naming Game".

Algorithm 1 Minimal Naming Game

procedure NAMING GAME
  function HEAR(word)
    if $word$ inside $Lexicon$ **then**
      $Lexicon.clear()$
      $Lexicon.add(word)$
      return $true$ ▷ Successful Game
    else
      $Lexicon.add(word)$
      return $false$ ▷ Unsuccessful Game
  function SPEAK
    if $Lexicon == \emptyset$ **then**
      $word \leftarrow generateRandomWord()$
    else
      $word \leftarrow Lexicon[rand()]$
    $Broadcast(word)$

Algorithm 2 Aggregation with Naming Game.

tlop

```
procedure AGGREGATION
    function STEP
        if SenseObstacle() then
            RandomTurn()
        if state = WALK then
            W ← Receive()
            count ← 1 //account for the bot itself
            for all w ∈ W do
                if Hear(w) then                ▷ Successful Game
                    count ← count + 1          ▷ increment
                else                           ▷ Unsuccessful Game
                    count ← 1                  ▷ reset
            P_Stay ← min(P_Base * count, 1)
            if random() < P_Stay then
                state ← STAY
                turns ← T_STAY
                Speak()
            else
                MoveStraight()
        else if state = STAY then
            turns ← turns − 1
            if turns = 0 then
                P_Stay ← min(P_Base * count, 1)
                if random() < P_Stay then
                    turns ← T_STAY
                else
                    state ← LEAVE
                    turns ← T_LEAVE
                    StopBroadcast()
        else if state = LEAVE then
            turns ← turns − 1
            if turns = 0 then
                state ← WALK
                MoveStraight()
```

To this end, we implemented our algorithm on a simulated version of the foot-bot, a version of the MarXbot [24] which is a differential wheeled robot. Besides the wheels, the foot-bots involved in the experiments were equipped with IR sensors in order to detect obstacles and with the range and bearing module developed for the swarmanoid project [25]: a combination of four IR sensors permanently rotating to make $360°$ scans and allowing to exchange data via IR light. This module is very suitable for the setting of this work as infra-red communication suffices to emulate the interactions between cockroaches [18, 19] and the possibility to exchange (a limited amount of) data is useful to play Language Games. We studied the behaviour of these robots under our model by running several experiments in a **6x7.5m arena** on the ARGoS simulator [26].

The experimental design that we used is described in Table 1. We varied only two parameters: the population size N and the P_{Base} probability. Moreover, the experiments were run with two versions of our algorithm: Vanilla (game-less, as presented in Subsection 2.1) and NG (with Naming Game, as presented in Subsection 2.2). As T_{STAY} and T_{LEAVE} obviously influence alignment/dispersion, their values were set, after some parameter tuning, to (resp.) 200 and 50 in all experiments. We performed 20 independent runs for each parameter configuration in order to have statistically meaningful results.

Table 1 Table of Experiments

Version	Label	NG?	N	P_{Base}	Runs
Vanilla	s20p14	No	20	0.14	20
	s20p17	No	20	0.17	20
	s20p20	No	20	0.20	20
	s100p14	No	100	0.14	20
	s100p17	No	100	0.17	20
	s100p20	No	100	0.20	20
NG	s20p14	Yes	20	0.14	20
	s20p17	Yes	20	0.17	20
	s20p20	Yes	20	0.20	20
	s100p14	Yes	100	0.14	20
	s100p17	Yes	100	0.17	20
	s100p20	Yes	100	0.20	20

The experiments were stopped when the swarm stabilised. The condition we used to determine whether stabilisation had happened or not was that every individual remain still for more than 600 time-steps. With T_{STAY} set at 200, it means that **every** robot has decided to stay three times in a row. However, it is possible that stabilisation never happens, in which case the experiments were stopped after 1 000 000 time-steps (100 000 seconds).

To study the impact the Naming Game and an aggregation behaviour have on each other, we manually analysed the final state of each run and we computed two quantities: the amount of aggregates and the amount of words in the game. Obviously, to evaluate the former quantity, one has to define an aggregate. For the purpose of this analysis, we viewed the final state as a graph where each node is a robot and distances lesser than the range and bearing's reach are edges. With these notions, we defined an aggregate as a set of nodes where each node is linked to any other node by at least two paths. With this definition, elongated aggregates are possible but chains of robots cannot bridge aggregates together nor can be considered aggregates by themselves. Moreover, if an experiment stopped because of the time constraint and if less than 90% of the robots were part of aggregates, the aggregates were not counted as they were deemed unstable. The quantity of words in the game is much simpler to count as it is straightforwardly the size of the set of all the single-word lexicons.

Finally, following [22], we computed the dispersion of the aggregates using the second moment, or variation, of the robots' positions. Using p_i as the position of robot i (among n in the aggregate), and $\bar{p}$ the centroid of these positions,

$$\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p_i \tag{1}$$

the second moment of the robots is given by:

$$v = \frac{1}{4r^2 n}\sum_{i=1}^{n} ||p_i - \bar{p}||^2 \tag{2}$$

where r is the radius of a robot. As the robots are not mere points and occupy space (wherein other robots can not fit), $4r^2$ normalises v to render it independent of r. As aggregates may have different sizes, we also added n in the denominator, which normalises v with regard to the size of the aggregate.

4. RESULTS

The data retrieved from our experiments are presented in Fig. 2. These plots were computed from our counts of the quantities of aggregates and words for each run of each experiment. The dots represent the mean number of aggregates/words by experiment and the error bars represent the standard deviation. Consequently, a shorter bar indicates that the final outcome of an experiment is consistent. Longer bars indicate that the outcome vary more widely.

The first observation to make is that, in small populations ($N = 20$), low values of P_{Base} are often insufficient to secure the emergence of aggregates. Nevertheless, with a sufficiently high value of P_{Base} (higher than 0.17), the NG version forms a single aggregate with a quorum of 90%.
We should note that, as the mean quantity of aggregates remains near one, the Vanilla version has a better success rate when P_{Base} is lower or equal to 0.17. However, as the count of the words shows, the NG-swarms consistently converge on a single word. Therefore, as explained in Section 2, they will eventually behave identically to the Vanilla-swarms. This means that, given more time, they would reach the same success rate. In any case, with the NG algorithm, a swarm of 20 robots *displays a normal aggregation behaviour* as the impact of the stay probability on the efficiency of aggregation is well-known [21].

However, with a larger population ($N = 100$), the results with NG are more surprising. Interestingly, the mean quantity of aggregates neatly increases with P_{Base}. As the error bars show low deviations, this quantity is also relatively consistent from run to run. Thus, with the right value, traditional aggregation in a single aggregate is still possible but the algorithm also allows the experimenter to *divide the swarm in a configurable quantity of aggregates*.

Furthermore, we observe that, apart from the cases where no aggregation happened, the quantity of words in the game remains extremely close to the quantity of aggregates. This correlation is made even clearer with the visual examples that can be seen in Fig. 3. We can see that each aggregate has its own word–which can act as a label–as if different naming games were played in

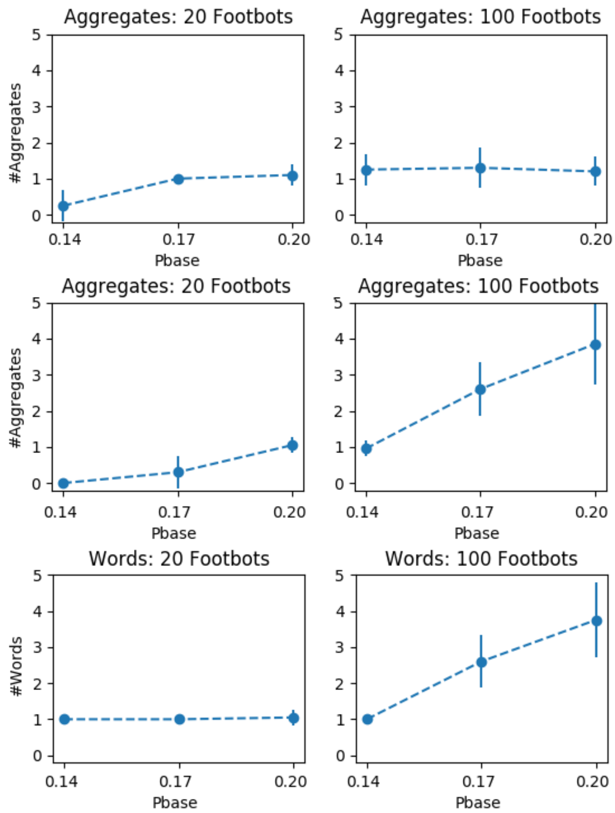

Fig. 2 Mean (dot) and standard deviation (bars) of the quantity of aggregates in vanilla/NG aggregation (resp. top/middle) and of words (bottom) in stabilised swarms of 20/100 (resp. left/right) robots or after 100 000 seconds. These charts show that the Vanilla algorithm displays a normal aggregation behaviour as, with appropriate P_{Base} [21], the robots consistently gather themselves in a single aggregate. Moreover, we observe a visual correlation between the final quantity of aggregates and the final quantity of words in the populations playing a Naming Game. Finally, in cases where the swarm does not stabilises (0 aggregates), the whole population still converges on one word.

each aggregate. This is incidentally also the case in small swarms but less ostensibly as they end up with one or zero aggregates and a single word. Consequently, the NG algorithm also *labels each robot of an aggregate with an identical word*.

Finally, looking at the average dispersion of the aggregate (or of the whole population in cases no aggregate formed) in Table 2, we can see that, in small populations ($N = 20$) with Vanilla, the variation of the positions of the robots (as computed from Eqs. (1) and (2)) is lower than the equivalent experiments with NG, which means that the aggregates are less dispersed with Vanilla than with NG. However, this difference is only significant with the *s20p17* experiment, which can easily be explained by having a look back at Fig. 2. Indeed, as the proximity of the mean quantity of aggregates to zero shows, the NG version of *s20p17* often failed to aggregate in this setting.

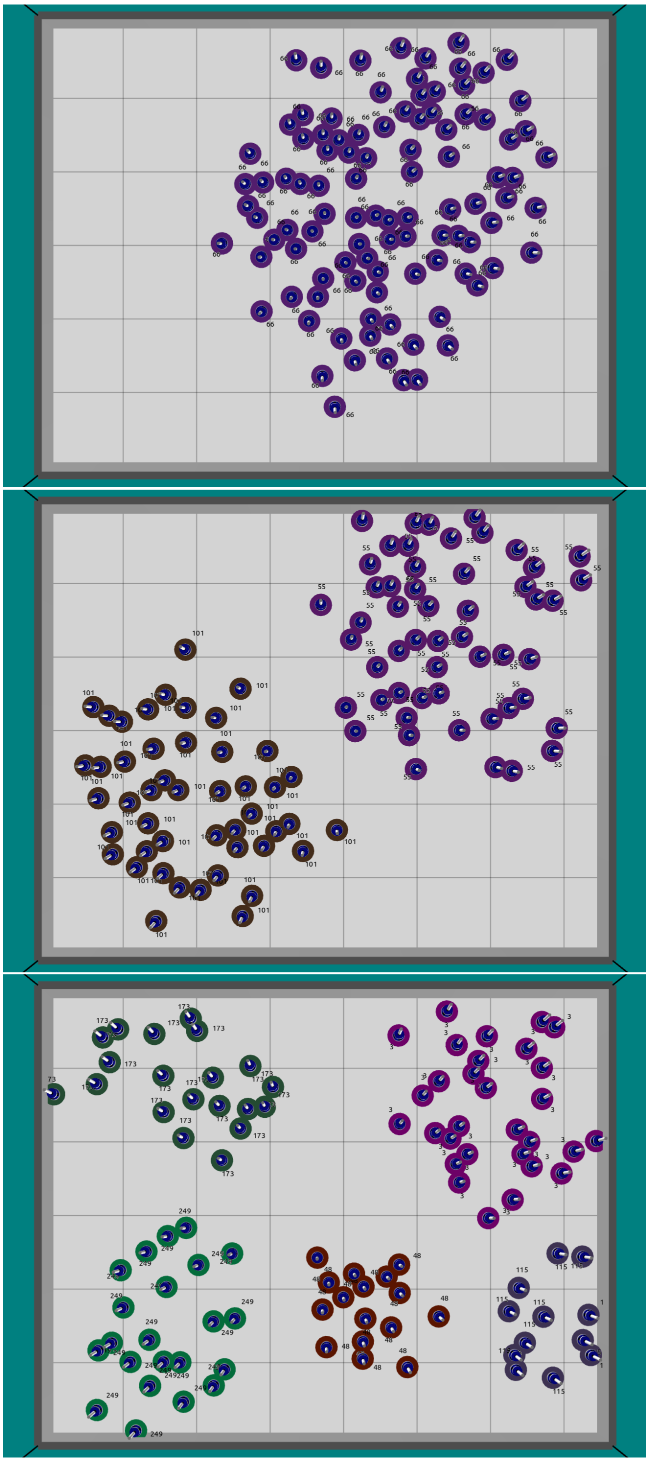

Fig. 3 Examples of stabilised swarms with a $N = 100$ and P_{Base} configured as (resp.) 0.14 (top), 0.17 $(middle)$, and 0.20 $(bottom)$. For visibility, each word is associated with a different colour in this display. We see that the quantity of aggregates increases with P_{Base} and that each aggregate converged on a different word.

Consequently, the average variation of the positions of the robots in the *s20p17* experiment with NG presented in Table 2 is increased by the dispersion of non-aggregated swarms. As the equivalent for Vanilla always aggregated, the variation is obviously much lesser.

Table 2 Average 2^{nd} moment of aggregates

	Vanilla	NG
s20p14	70.3017	85.6672
s20p17	12.5599	56.6773
s20p20	16.4397	17.8883
s100p14	65.0273	37.1738
s100p17	80.9566	19.9819
s100p20	85.5426	18.1084

Yet, in large populations ($N = 100$) the roles are reversed and NG is now much less dispersed. The explanation here is more subtle: As the density of robots is quite high (N increased but the size of the arena remained the same), many are within communication reach from the initialisation of the runs. Consequently, in the Vanilla version, many robots stop almost immediately and, as they are already well surrounded, remain in this state indefinitely. The scarce robot that did not stop immediately thus quickly meets the aggregate (which spans the whole arena) and enters the $STAY$ state too. Thus the aggregation process is fast but non-qualitative. In the NG version, however, the robots cannot agree on a word at the outset. They are therefore **forced** too explore the arena and aggregate much later, but in tighter aggregates. Thus, *the aggregation is slower but qualitative*.

To summarise, the influence of the Naming Game on aggregation is minor with small populations ($N = 20$) but provides remarkable benefits with larger population sizes ($N = 100$) as the swarm *can* then *divide itself in several, tighter, aggregates*. However, more experiments with more population sizes are needed to fully understand the connection between the naming game and the multiplication of aggregates. For instance, it is not clear here whether the density or the population size is the main cause of this quantitative change in behaviour. Conversely, the impact of aggregation on the Naming Game is that, by reducing robot mobility through aggregation, different naming games are played in parallel, thus preventing the population from converging to a single word.

5. CONCLUSIONS

We believe that evolutionary linguistic models such as the naming game can offer robotic swarms a way to make collective decisions when several unforeseen alternatives are available. This paper concentrated on implementing such a model in an aggregation behaviour, which can be seen as a prerequisite for other types of collective behaviours and tasks [1].

The results of our simulations with the ARGoS simulator show that, in small population sizes ($N = 20$), the aggregation dynamics with the naming game remains unchanged with respect to a vanilla aggregation algorithm without the naming game, but with the added feature of the convergence on a single word, which could be used later in other collective behaviours. However, with larger population sizes ($N = 100$), the swarm displays three interesting emerging features: (a) The possibility to select the quantity of final aggregates by changing the value of P_{Base}; (b) more compact aggregates; (c) aggregates associated each to a different word (labelling).

Our future work will be concerned with performing experiments in more environmental conditions to fully characterize the dynamics.We will introduce more variations in the parameters, both to confirm already visible trends and to study their scalability. Finally, we will continue to study the Naming Game in various swarm robotics settings in order to study how this collective decision-making can have practical applications in unforeseen situations. According to our agenda, our plan is to study the interplay between evolutionary linguistic models and different collective behaviours in swarm robotics in order to (a) understand whether the collective behaviour qualitative or quantitative performance can be improved (like in the case presented here) and (b) study the dynamics of evolutionary linguistic models in novel realistic situations as opposed to the idealistic ones considered in the classical models.

6. ACKNOWLEDGEMENTS

This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program Investments for the future managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02)

REFERENCES

[1] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[2] S. Nolfi and D. Floreano, *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT press, 2000.

[3] V. Trianni, *Evolutionary swarm robotics: evolving self-organising behaviours in groups of autonomous robots*. Springer, 2008, vol. 108.

[4] D. Floreano, S. Mitri, S. Magnenat, and L. Keller, "Evolutionary conditions for the emergence of communication in robots," *Current biology*, vol. 17, no. 6, pp. 514–519, 2007.

[5] J. Hurford, C. Knight, and M. Studdert-Kennedy, "Evolution of human language," 1998.

[6] L. Steels, "Emergent adaptive lexicons," 1996.

[7] L. Steels, "A self-organizing spatial vocabulary," *Artificial life*, vol. 2, no. 3, pp. 319–332, 1995.

[8] L. Steels and M. Loetzsch, "The grounded naming game," *Experiments in cultural language evolution*, vol. 3, pp. 41–59, 2012.

[9] A. Baronchelli, M. Felici, V. Loreto, E. Caglioti, and L. Steels, "Sharp transition towards shared vocabularies in multi-agent systems," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 06, p. P06014, 2006.

[10] A. Baronchelli, "Role of feedback and broadcasting in the naming game," *Physical Review E*, vol. 83, no. 4, p. 046103, 2011.

[11] G. Valentini, E. Ferrante, H. Hamann, and M. Dorigo, "Collective decision with 100 kilobots: speed versus accuracy in binary discrimination problems," *Autonomous Agents and Multi-Agent Systems*, pp. 1–28, 2015. [Online]. Available: `http://dx.doi.org/10.1007/s10458-015-9323-3`

[12] G. Valentini, E. Ferrante, and M. Dorigo, "The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives," *Frontiers in Robotics and AI*, vol. 4, p. 9, 2017.

[13] V. Trianni, D. De Simone, A. Reina, and A. Baronchelli, "Emergence of consensus in a multi-robot network: from abstract models to empirical validation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 348–353, 2016.

[14] M. Rubenstein, C. Ahler, and 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.

[15] 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," *Autonomous Robots*, vol. 17, no. 2, pp. 223–245, 2004.

[16] J.-L. Deneubourg, A. Lioni, and C. Detrain, "Dynamics of aggregation and emergence of cooperation," *The Biological Bulletin*, vol. 202, no. 3, pp. 262–267, 2002.

[17] R. Jeanson, C. Rivault, J.-L. Deneubourg, S. Blanco, R. Fournier, C. Jost, and G. Theraulaz, "Self-organized aggregation in cockroaches," *Animal behaviour*, vol. 69, no. 1, pp. 169–180, 2005.

[18] S. Garnier, C. Jost, R. Jeanson, J. Gautrais, M. Asadpour, G. Caprari, and G. Theraulaz, "Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots," in *ECAL*, vol. 3630. Springer, 2005, pp. 169–178.

[19] N. Correll and A. Martinoli, "Modeling self-organized aggregation in a swarm of miniature robots," in *IEEE 2007 International Conference on Robotics and Automation Workshop on Collective Behaviors inspired by Biological and Biochemical Systems*, no. SWIS-CONF-2007-002, 2007.

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

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

[22] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß, "Self-organized aggregation without computation," *The International Journal of Robotics Research*,

vol. 33, no. 8, pp. 1145–1161, 2014.

[23] V. Loreto, A. Baronchelli, and A. Puglisi, "Mathematical modeling of language games," in *Evolution of communication and language in embodied agents*. Springer, 2010, pp. 263–281.

[24] M. Bonani, V. Longchamp, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada, "The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 4187–4193.

[25] 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 Robotics & Automation Magazine*, vol. 20, no. 4, pp. 60–71, 2013.

[26] C. Pinciroli, V. Trianni, R. O´Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle *et al.*, "Argos: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm intelligence*, vol. 6, no. 4, pp. 271–295, 2012.