Complexity Measures in Automatic Design of Robot Swarms: An Exploratory Study

Andrea Roli[1(✉)] , Antoine Ligot[2] , and Mauro Birattari[2]

[1] Department of Computer Science and Engineering, Campus of Cesena,
Alma Mater Studiorum, Università di Bologna, Cesena, Italy
andrea.roli@unibo.it
[2] IRIDIA, Université libre de Bruxelles, Brussels, Belgium

Abstract. The design of control software for robot swarms is a challenging endeavour as swarm behaviour is the outcome of the entangled interplay between the dynamics of the individual robots and the interactions among them. Automatic design techniques are a promising alternative to classic *ad-hoc* design procedures and are especially suited to deal with the inherent complexity of swarm behaviours. In an automatic method, the design problem is cast into an optimisation problem: the solution space comprises instances of control software and an optimisation algorithm is applied to tune the free parameters of the architecture. Recently, some information theory and complexity theory measures have been proposed for the analysis of the behaviour of single autonomous agents; a similar approach may be fruitfully applied also to swarms of robots. In this work, we present a preliminary study on the applicability of complexity measures to robot swarm dynamics. The aim of this investigation is to compare and analyse prominent complexity measures when applied to data collected during the time evolution of a robot swarm, performing a simple stationary task. Although preliminary, the results of this study enable us to state that the complexity measures we used are able to capture relevant features of robot swarm dynamics and to identify typical patterns in swarm behaviour.

1 Introduction

The behaviour of a swarm of robots is the result of the dynamic interplay among the robots, and between robots and environment. As a consequence, the design of control software for a robot swarm presents hard challenges. Typical techniques for designing robot swarm are based on code-and-fix methods [4], usually tailored to the specific problem at hand. A promising alternative to these *ad hoc* approaches is provided by automatic design techniques [9], which are especially suited to deal with the inherent complexity of swarm behaviours. In automatic methods, the design problem is cast into an optimisation problem, whereby the solution space contains instances of control software and an optimisation algorithm is applied to tune the free parameters of the architecture [10,29]. For the

sake of completeness, we observe that the design process is not simply reduced to an optimisation problem because it also involves the definition of proper merit factors and experimental settings, likewise learning methods.

Recently, some information-theoretical measures have been proposed for the analysis and design of the behaviour of single autonomous agents [1,8,25,28]. These studies support the use of information theory and complexity science concepts in the field of autonomous agents and robotics. We believe that these techniques may be fruitfully applied also to swarms of robots. Indeed, complex systems science may provide a corpus of theories and methods that enable the designer to formally and quantitatively analyse the dynamics of a robot swarm and its internal information processes.

Complexity measures may be applied to the automatic design of robot swarms with the following objectives:

1. understanding individual and swarm behaviour from observations of measurable quantities (e.g. sensor readings, actuation, controller state);
2. providing task-agnostic merit factors for the automatic design procedures;
3. classifying swarm tasks in terms of their intrinsic complexity so as to optimally tune the complexity of individual robot and robot interactions.[1]

In the long term we plan to address the following questions: *(i)* Are the intuition behind the measures in accordance with the observed robot swarm behaviour? And is the observed behaviour coherent with the complexity values measured? *(ii)* What are the most informative measures? *(iii)* What are the complexity measures most suited for such an application? *(iv)* Are there phenomena in the swarm behaviour that can be detected just by observing the complexity values measured? The outcome of this study is expected to provide guidelines for the choice of the most informative indicators for more complex tasks.

In this work, we present a preliminary study on the applicability of complexity measures to robot swarm dynamics. The aim of this investigation is to compare and analyse prominent complexity measures when applied to data collected during the time evolution of a robot swarm performing a simple stationary task. In the following, we first summarise the measures considered in this study in Sect. 2; subsequently, we detail the robot swarm task (Sect. 3). In Sect. 4, we provide a summary of the main results, emphasising the ones that enable us contributing to answer the questions raised above. Finally, we conclude with an outlook of ongoing and future work.

2   Measures of Complexity

In the scientific literature the word *complexity* is overloaded, as it appears with different meanings, each related to a specific interpretation of the term. As a consequence, there is no unique measure of complexity and in fact many metrics have been proposed in the literature. In general, each metric addresses one

---

[1] This goal is motivated by a conjecture on the so-called reality gap, which has been advanced in [3,10].

specific aspect of the general notion of complexity, therefore we should aim at producing a *complexity fingerprint* by evaluating several measures, rather than identifying a single metric able to summarise all the relevant properties related to complexity.

A measure of complexity was first proposed by Kolmogorov [14] who provided an algorithmic view of information: the complexity of a string of symbols is defined as the length of the shortest program producing it. As this measure is not computable some approximations have been proposed, such as the ones based on compression algorithms. In fact, algorithmic complexity estimates the amount of randomness in a string, as they turn out to be very low for regular sequences and maximised for completely random strings. The definition of complexity we are interested in tries to capture the notion of presence of (dynamical) patterns, often related to the extent to which correlations distribute across the element of the system observed [13]. Intuitively, high complexity is associated to situations between order and disorder, as patterns in both ordered and completely random dynamics are negligible. Along this line, several measures have been proposed [12, 13, 16, 18, 26]. However, a survey on the literature on complexity measures is out of the scope of this contribution and we refer the interested reader to prominent works on the subject [2, 13, 18, 20, 24]. A nice introduction to information theory for complex systems can be found in the lecture notes by Lindgren [17].

In this work, we focus primarily on the complexity of the dynamics of the system observed in its environment, rather than the individual complexity of a controller of an isolated robot. Moreover, as a consequence of the fact that we deal with data collected during experiments, the measures used should be applied to time series of finite length. Among the measures proposed in the literature, we selected and implemented the following ones:

1. Shannon entropy [27]
2. Block entropy and entropy excess [22]
3. Correlation information [17]
4. Mutual information [6]
5. LMC complexity [21]
6. Lempel-Ziv complexity [15]
7. *bz2* compression factor [5]
8. Linguistic complexity [30]
9. Set-based complexity [11]

The choice of these metrics has been motivated by the intent of covering the diverse facets of complexity, and also taking into account computational requirements.[2]

Measures 1–5 are based on the Shannon entropy of a sequence s of symbols in a finite set $\mathcal{X}$. We suppose that the frequency of symbols appearing in s approximates the probability distributions of the symbols. Therefore, we can

---

[2] Indeed, due to excessive computational resources required, for this preliminary step we did not applied measures of complexity based on model construction, such as the ones by Crutchfield et al. [7].

provide the definition of entropy in terms of random variables. Let X be a random variable which can assume values from a finite and discrete domain $\mathcal{X}$, the Shannon entropy of X is defined as

$$H(X) = - \sum_{x \in \mathcal{X}} P(x) \ log \ P(x)$$

where the logarithm is expressed in base 2. This definition can be extended to blocks of symbols of length n in s, so as to take into account also correlations among symbols. This leads to the definition of the *block entropy* of length n:

$$H_n = - \sum_{s_n \in \mathcal{S}} P(s_n) \ log \ P(s_n)$$

The *entropy excess*[3] is defined as the difference between block entropies of length n and $n-1$ and estimates the information required to predict the n-th symbol conditioned to the observation of $n-1$ preceding symbols. In formulas:

$$h_n = \Delta H_n = H_n - H_{n-1}$$

We can extend this process to the second derivative (in discrete domains) and obtain the *correlation information* from length n:

$$k_n = \Delta^2 H(n) = -H(n) + 2H(n-1) - H(n-2), \ n \geq 3$$

Intuitively, the peaks of k_n identify significant block regularities, i.e. maximum gain in information for specific block lengths.

Also the *mutual information $I(X,Y)$* between random variables X and Y is defined in terms of entropies and estimates the average information one gains about Y after the observation of X, and viceversa:

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

where $H(X,Y)$ denotes the conjunct entropy of X and Y.

For completeness, we also introduce the LMC complexity[4] which is defined in terms of entropy and disequilibrium:

$$LMC(X) = H(X) \cdot D(X)$$

where $D(X) = \sum_{x \in \mathcal{X}} \left(P(x) - \frac{1}{|\mathcal{X}|} \right)^2$. Unfortunately, this metric is quite sensitive to numeric factors—mainly the values of H and D at the borders—and the results it returns should be taken with care.

Measures 6–9 are instead based on computing properties of the sequence at hand, rather than referring to a probability distribution. In particular, the

---

[3] Not to be confused with the *excess entropy* [26], which is defined for $n \to \infty$.
[4] The name comes from the name initials of its inventors.

Lempel-Ziv complexity (LZ) is a sort of algorithmic information measure computable on finite sequences, therefore it estimates the randomness of a string. $LZ(s)$ returns the number of shortest different blocks composing s. Along the same line is the compression factor achieved when compressing the string, in our case with algorithm *bzip*, which takes into account blocks of different size. *Linguistic complexity* is another metric that based on the occurrences of different blocks in a sequence of symbols and is computed for blocks of varying size.

Finally, the complexity of a set of strings $S = \{s_1, s_2, \ldots, s_N\}$ can be estimated by means of the *set-based complexity SBC*, which accounts for the informative contribution of each string to the set. The intuition behind this measure is that a random string and a duplicated string do not contribute to the overall complexity of the set. This metric is defined in terms of Kolmogorov complexity $K(s)$ and it is empirically computed by approximating it with a compression algorithm, providing an estimation $\hat{K}(s)$. Based on algorithmic complexity, the *distance* between two strings can be computed as follows:

$$d(i,j) = \frac{\hat{K}(x \oplus y) - min(\hat{K}(x), \hat{K}(y))}{max(\hat{K}(x), \hat{K}(y))}$$

where $x \oplus y$ denotes the concatenation of strings x and y. The SBC of the set of strings S is defined as:

$$SBC(S) = \sum_{i=1}^{N} \hat{K}(s_i)F_i(S)$$

where

$$F_i(S) = \frac{2}{N(N-1)} \sum_{j \in \{1,2,\ldots,N\}, i \neq j} d_{ij}(1 - d_{ij}), \ \ d_{ij} := d(s_i, s_j)$$

3   Case Study: Random Walk with Collision Avoidance

We defined a case study that requires a simple software controller for the robots and few parameters to be tuned. Moreover, the mission the swarm has to accomplish should be modelled as a stationary process, and its level of complexity should be sufficiently high to be measured and produce non-trivial results. At the same time, the complexity should be limited so as to allow an easy interpretation of the results. We performed our experiments in a simulated environment by the means of ARGoS [23], one of the most widespread swarm robotics simulators. The robot chosen to be simulated is an *e-puck*, equipped with 8 infra-red proximity sensors positioned around the circular body and two wheels.

3.1   Behaviour: Random Walk with Collision Avoidance

The random walk behaviour is a strategy for space exploration commonly used in swarm robotics. We implemented this strategy as the alternate execution of

Function `ControlStep()`

    if *State == STRAIGHT* **then**
        if *StraigthSteps == 0 OR obstacle in front* **then**
            State = LEFT or RIGHT with same probabilities;
            Bernouilli(0,5) == 1 ? State ← LEFT : State ← RIGHT;
            TurningSteps ← Uniform(0, $\frac{R_a}{10}$);
        else
            `GoStraight();`
            StraightSteps ← StraightSteps - 1;
        end
    end
    if *State != STRAIGHT* **then**
        if *TurningSteps > 0* **then**
            if *State == RIGHT* **then**
                `TurnRight();`
            else
                `TurnLeft();`
            end
            TurningSteps ← TurningSteps - 1;
        else
            State ← STRAIGHT;
            StraightSteps ← W_s;
            `ControlStep();`
        end
    end

Algorithm 1. Control step of the random walk behaviour executed every 100 milliseconds. The methods GoStraight(), TurnRight() and TurnLeft() are responsible for affecting the required values to the wheels actuator in order for the robot to move forward, rotate clockwise or anti-clockwise, respectively. The recursive call to ControlStep() allows the robot to verify the absence of obstacle before starting to move forward.

straight movements and static rotations: at each time step of an experiment, the e-puck robots can either move forward for a given distance or rotate at a given angle. In our implementation of the random walk behaviour the robots walk straight for a maximal distance W_s. After this maximal distance is travelled, or if an obstacle is perceived in front of the robot, the static rotation phase is triggered. During the rotation, a robot turns left or right with same probability, with a rotation angle given by a multiple of 10° taken uniformly between 0 and a maximal angle R_a. Once the robot has completed the rotation, it can once again move forward under the condition that no obstacles are on the way. Conversely, if the path is not clear in front of the robot, another static rotation phase is immediately started. Algorithm 1 resumes the behaviour that we implemented.

3.2 Experimental Settings

For this study, we executed multiple runs of the random walk behaviour with two parameters: the maximal straight distance $W_s \in \{10, 20, 30\}$ expressed in centimeters, and the maximal angle of rotation $R_a \in \{40, 90, 180\}$ expressed in degrees.
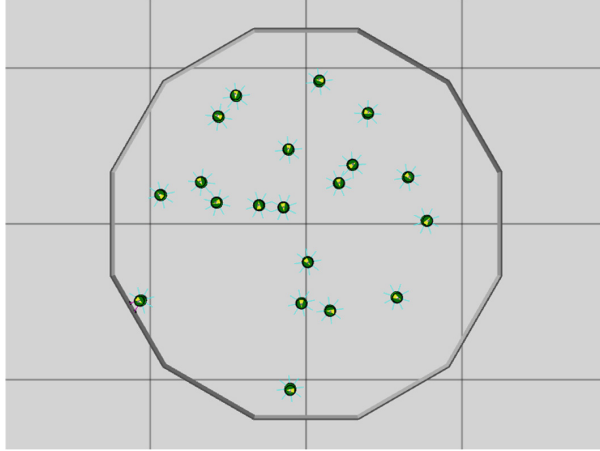

Fig. 1. Picture of the enclosed environment setup containing a swarm of 20 e-puck robots. The 8 cyan lines around each robots represent their proximity sensors. (Color figure online)

We ran two types of experiments. The first one is a control scenario involving a single robot that moves in an infinite space with no obstacles nor boundaries. This scenario represents a baseline for the comparisons with the swarms. The second experiment involves a swarm of $N \in \{1, 10, 20, 40\}$ robots moving in an enclosed environment in which the walls form a dodecagonal shape with an area equal to $4.91 \, \mathrm{m}^2$ (see Fig. 1). The swarm is composed of robots all controlled by the same random walk behaviour. At the beginning of each experiment, the robots are uniformly distributed in the dodecagonal arena. Every possible combinations of the parameters W_s and R_a were used in the two experiments. Each experiment was repeated 10 times. Therefore, a total of 450 experiments were ran.

The state of a robot performing this kind of random walk can be simplified and expressed by means of three possible states: Straight, Left, and Right. Hence, at each instant, the state of the whole swarm of N robots can be represented by a vector of symbols, each from the alphabet $\{S, L, R\}$. For each run, we recorded the state vector of the swarm every 100 ms. As runs last 20 min, a total of 12000 state vectors were recorded for each experiment. The complexity measures were applied to this symbolic sequence depending on the definition of the measure, i.e.,

either to the whole vector state (e.g., for set-based complexity) or by averaging the values computed across all the robots (e.g., for entropies).

4   Results

The factors influencing swarm behaviour that we expect to be reflected into a complexity metrics analysis are the number of forward steps, the maximum turning angle and the number of robots in the arena. In particular, the metrics should provide information on the amount of regularity in robots' trajectories and on their interactions. As we will show, although preliminary, the results of this analysis enable us to state that the complexity measures we used are able to capture these relevant features of robot swarm dynamics. Moreover, we discovered that some metrics were able to capture non-trivial properties of the dynamics of the swarm. In this paper we show and discuss a representative sample of the results. The metrics we have omitted in this discussion are anyway in agreement with the ones we have chosen for this presentation.

In the following plots, colours are used to differentiate among the three possible turning angle values: 40° in red, 90° in green and 180° in blue. The plots shown are produced by analysing one run for each possible combination of experimental factors; qualitatively analogous results are observed in the other runs.
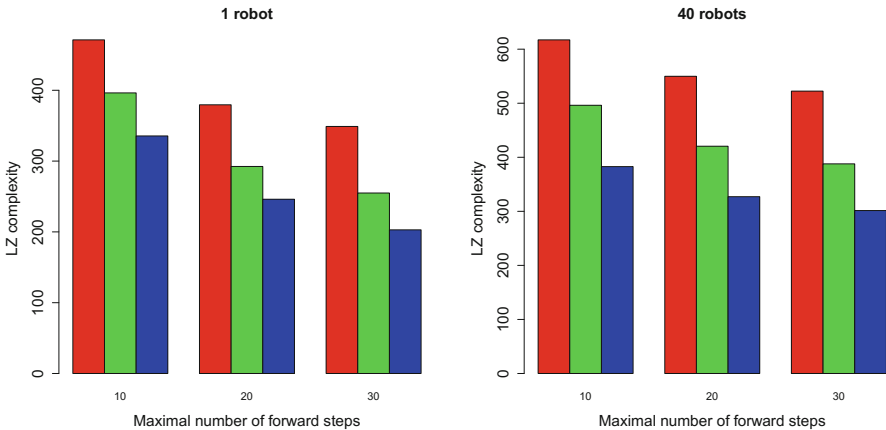

Fig. 2. LZ complexity for the case with 1 robot (left) and 40 robots (right) in the dodecagonal arena. Maximum turning angles: ■ 40° ■ 90° ■ 180° (Color figure online)

In general, Shannon entropy and all the metrics measuring randomness are in agreement with the expectations, as they show that randomness increases if the number of forward steps decreases, the maximum turning angle decreases or the number of robot increases. In Fig. 2 a representative example is shown for the LZ complexity. Note that the maximum values reached in the case of

40 robots are higher than those for one robot, providing a quantitative account of the positive correlation between number of robots and randomness in their trajectories. In addition, the LZ complexity decreases with the number of steps and the maximum turning angle, specifically confirming that robots' trajectories are more regular when they have more possibilities to avoid obstacles.
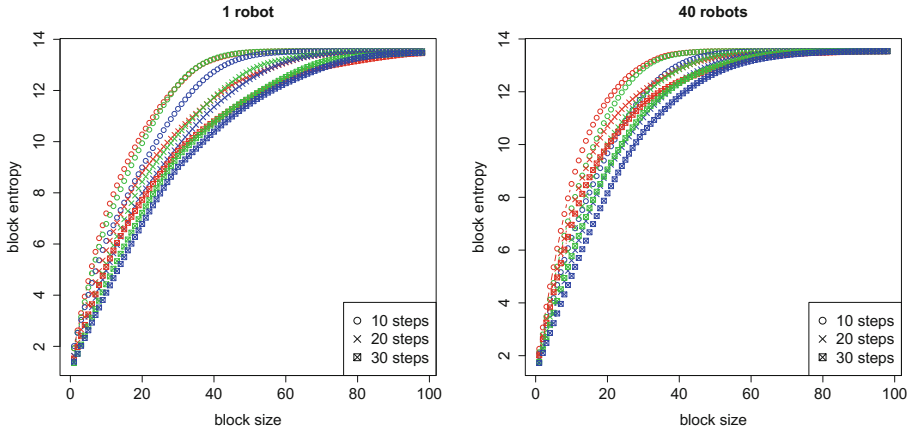

Fig. 3. Block entropy for the dodecagonal arena case, with 1 robot (left) and the 40 robots (right). Maximum turning angles: ■ 40° ■ 90° ■ 180°

Block entropies and their derivatives are particularly informative because they provide a picture of the correlations at different lengths in the dynamics of the swarm. The block entropy as a function of block size is plotted in Fig. 3 for the two extreme cases of the scenario with the dodecagonal arena. As expected, the curves grow more rapidly for the dynamics characterised by a higher level of randomness. The curves saturate when the length of the block considered is about 40; in fact, as data series are of finite length, the frequency of large blocks is underestimated and the block entropy values tend to converge even if, in principle, the asymptote should have a strictly positive derivative for non-periodic dynamics [17]. Therefore, the block entropy values are meaningful for shorter block lengths. The block entropy trends suggest two main observations. First, the initial slope of the curves is higher on average in the 40 robots case; this is a direct consequence of the fact that the denser the robots the less regular their trajectories in the arena. Second, the top and bottom limiting curves correspond to the least (10 steps, 40°) and most (30 steps, 180°) regular case, respectively.

The correlation information (i.e. the second discrete derivative of the block entropy) makes it possible to identify the points at which the block entropy slope changes, thus providing a tool for a detailed inspection of the regularities in the time series. The plots in Fig. 4 summarise the results of the correlation length analysis. The most notable fact to observe is that in every condition considered, and independently of the turning angle, there is a marked peak corresponding to
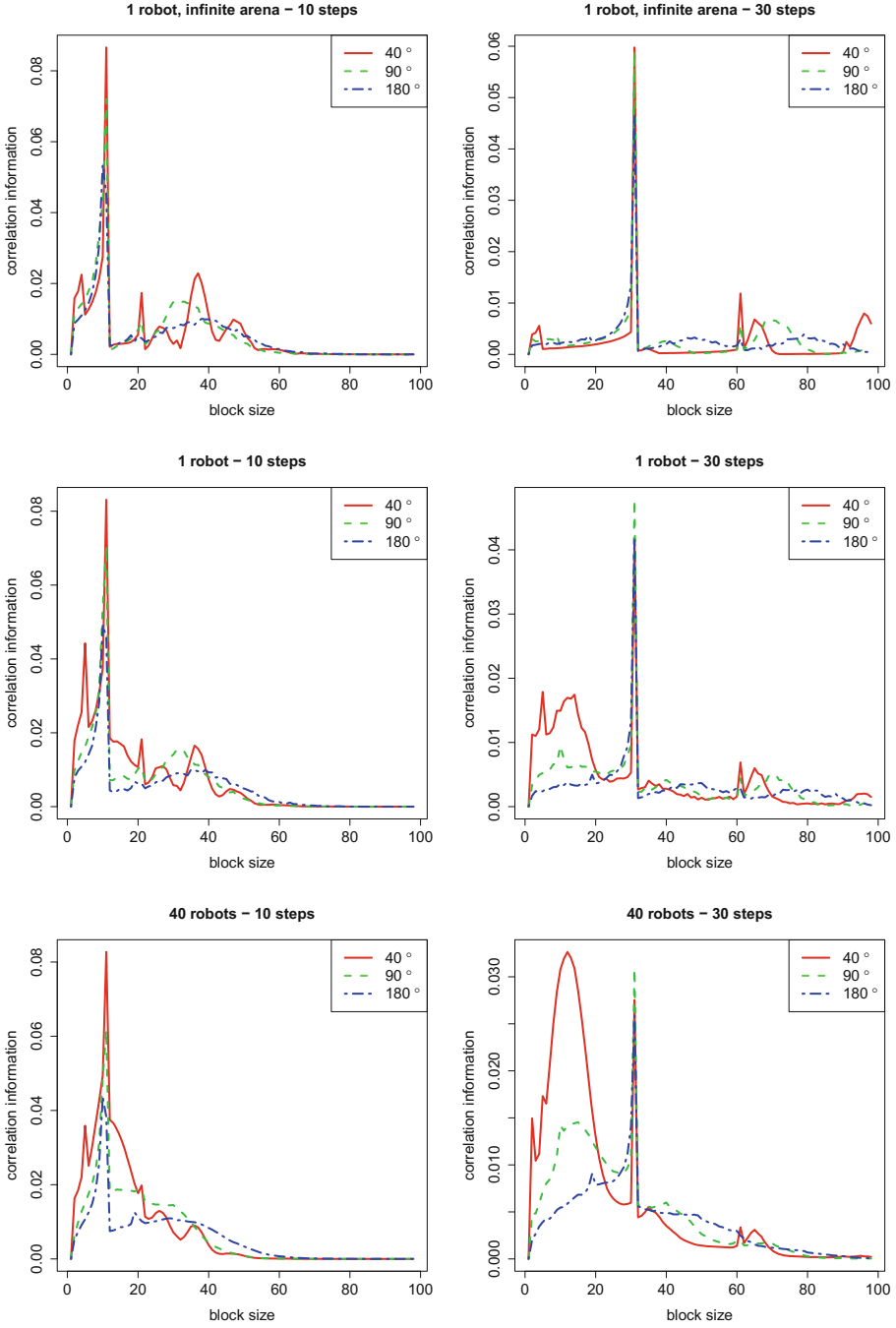
Fig. 4. Correlation information from length k, where k is the block length.

the number of forward steps. Indeed, this is one of the most relevant regularities in robots' trajectories. We can also observe lower peaks corresponding to multiples of the number of forward steps. This picture is particularly striking in the control case (one robot, infinite arena) and gets blurred when delimiting walls are present and mainly when robots in the arena are dense, as their avoidance behaviour introduces randomness in their trajectories. We observe also a surprising phenomenon: a second peak appears at the left of the previously mentioned one. This peak is particularly marked in the case of 40 robots and 30 forward steps, where it is even higher than the other peak. This second local maximum captures the pattern of turning moves of the robots trying to avoid an obstacle. Indeed, the location of this peak gives us an indication of the average number of turning moves the robots have to take before finding a free corridor to move ahead. Whilst this phenomenon deserves a further in-depth investigation, this result is remarkable as it shows that correlation information provides a fine tool to detect—possibly unforeseen—regularities.
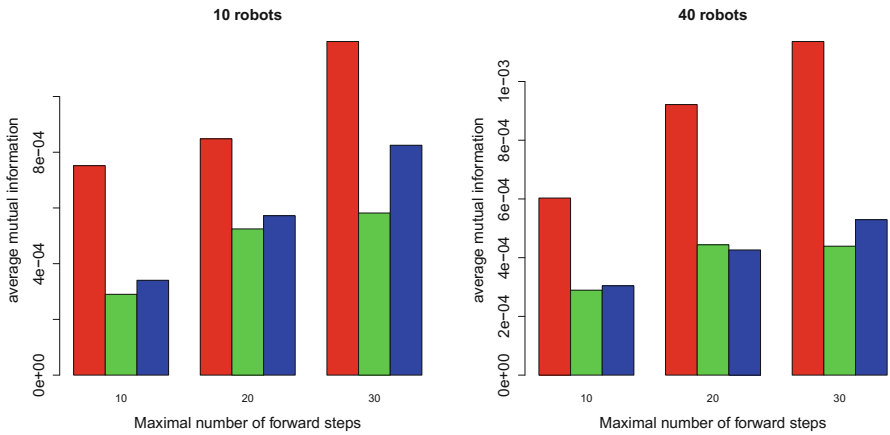

Fig. 5. Mutual information for the dodecagonal arena case, with 1 robot (left) and the 40 robots (right). Maximum turning angles: ■ 40° ■ 90° ■ 180°

A mutual information analysis of robots' trajectories may provide an estimation of the reciprocal influence between robots. Mutual information is computed for all the possible robot pairs and then averaged. The barplots in Fig. 5 show that the interdependence among robots is highest for the case of 40 robots and that the interactions are stronger for smaller turning angles. This analysis is in agreement with the expectations and complements the information gained by the previous metrics.

For completeness, we conclude this section by mentioning the results returned by the application of the set-based complexity. SBC is computed by considering the sequence of swarm states as a set of strings; therefore, it is a measure of the ensemble of robots, rather than of the robots taken individually. Barplots of
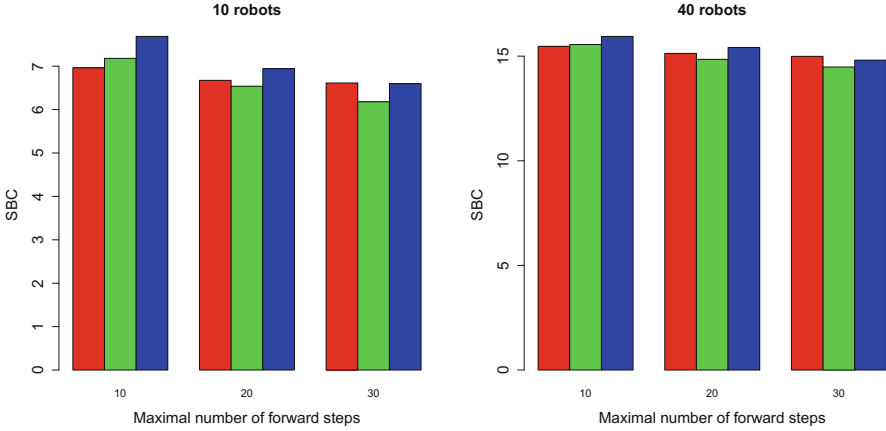
Fig. 6. Set-based complexity for the dodecagonal arena case, with 1 robot (left) and the 40 robots (right). Maximum turning angles: ■ 40° ■ 90° ■ 180°

this analysis are depicted in Fig. 6. We observe that SBC does not differentiate considerably as a function of forward steps and maximum turning angle. Conversely, it is worth to emphasise that the SBC values double moving from 10 to 40 robots, and also that the impact of forward steps number is stronger in the case of 10 robots, where the interference among robots is limited. Nevertheless, as the robots are mainly characterised by random walk, the potential of SBC can not be expressed completely and we expect that this metric could be particularly useful in non-stationary cases.

5   Conclusion and Future Work

The results of this exploratory study show that complexity metrics can capture relevant features, such as patterns, in traces of robot swarm dynamics. We have chosen the most known complexity measures, mainly from information theory, and applied them to a simple task for swarm robotics characterised by a stationary dynamics. As expected, metrics devised for measuring specific dynamical traits return similar results and an heterogeneous selection of them is likely to be the best choice to produce a *complexity fingerprint* of the system. A minimal fingerprint for a stationary case should be composed of metrics focusing on *(a)* randomness (e.g. LZ complexity), *(b)* patterns (e.g. block entropy and its derivatives) and *(c)* interdependence among robots in the swarm (e.g. mutual information).

We are currently enlarging the set of metrics, by including also statistical complexity measures based on model construction, and we plan to apply also local measures [19] and information theoretical measures specifically designed for capturing dynamical properties of the system [31]. Experiments on further stationary cases are planned, such as flocking and memoryless foraging with

random walk. The next step will be to address also non-stationary cases, like e.g. aggregation, so as to be able to tackle swarm missions in which robots may be characterised by changes in their dynamical behaviour. As stated in the introduction, our aim is to devise tools for helping the automatic design of controllers for robot swarms, so our research agenda include as a further step the use of complexity measures both as analysis tool and task-agnostic merit factors.

References

1. Ay, N., Bertschinger, N., Der, R., Güttler, F., Olbrich, E.: Predictive information and explorative behavior of autonomous robots. Eur. Phys. J. B - Condens. Matter Complex Syst. **63**(3), 329–339 (2008)
2. Badii, R., Politi, A.: Complexity: Hierarchical Structures and Scaling in Physics, vol. 6. Cambridge University Press, Cambridge (1999)
3. Birattari, M., Delhaisse, B., Francesca, G., Kerdoncuff, Y.: Observing the effects of overdesign in the automatic design of control software for robot swarms. In: Dorigo, M., Birattari, M., Li, X., López-Ibáñez, M., Ohkura, K., Pinciroli, C., Stützle, T. (eds.) ANTS 2016. LNCS, vol. 9882, pp. 149–160. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44427-7_13
4. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. Swarm Intell. **7**(1), 1–41 (2013)
5. http://www.bzip.org. Accessed 30 Nov 2016
6. Cover, T., Thomas, J.: Elements of Information Theory. Wiley, Hoboken (2012)
7. Crutchfield, J.: The calculi of emergence: computation, dynamics, and induction. Physica D **75**, 11–54 (1994)
8. Edlund, J., Chaumont, N., Hintze, A., Koch, C., Tononi, G., Adami, C.: Integrated information increases with fitness in the evolution of animats. PLoS Comput. Biol. **7**(10), e1002236 (2011)
9. Francesca, G., Birattari, M.: Automatic design of robot swarms: achievements and challenges. Front. Rob. AI **3**, 29 (2016)
10. Francesca, G., Brambilla, M., Brutschy, A., Trianni, V.: AutoMoDe: a novel approach to the automatic design of control software for robot swarms. Swarm Intell. **8**(2), 89–112 (2014)
11. Galas, D., Nykter, M., Carter, G., Price, N.: Biological information as set-based complexity. IEEE Trans. Inf. Theory **56**, 667–677 (2010)
12. Gell-Mann, M., Lloyd, S.: Information measures, effective complexity, and total information. Complexity **2**(1), 44–52 (1996)
13. Grassberger, P.: How to measure self-generated complexity. Phys. A: Stat. Mech. Appl. **140**(1–2), 319–325 (1986)
14. Kolmogorov, A.: Three approaches to the quantitative definition of information. Prob. Inf. Transm. **1**(1), 1–7 (1965)

15. Lempel, A., Ziv, J.: On the complexity of finite sequences. IEEE Trans. Inf. Theory **22**(1), 75–81 (1976)
16. Li, W.: On the relationship between complexity and entropy for Markov chains and regular languages. Complex Syst. **5**(4), 381–399 (1991)
17. Lindgren, K.: Information theory for complex systems - an information perspective on complexity in dynamical systems, physics, and chemistry. Chalmers (2014). http://studycas.com/c/courses/it
18. Lindgren, K., Nordahl, M.: Complexity measures and cellular automata. Complex Syst. **2**(4), 409–440 (1988)
19. Lizier, J.: The Local Information Dynamics of Distributed Computation in Complex Systems. Springer Theses Series. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-32952-4
20. Lloyd, S.: Measures of complexity: a nonexhaustive list. IEEE Control Syst. Mag. **21**(4), 7–8 (2001)
21. Lopez-Ruiz, R., Mancini, H., Calbet, X.: A statistical measure of complexity. Phys. Lett. A **209**, 321–326 (1995)
22. Nicolis, G., Nicolis, C.: Foundations of Complex Systems: Emergence, Information and Predicition. World Scientific, Singapore (2012)
23. Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L., Dorigo, M.: ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics. Swarm Intell. **6**(4), 271–295 (2012)
24. Prokopenko, M., Boschetti, F., Ryan, A.: An information-theoretic primer on complexity, self-organization, and emergence. Complexity **15**(1), 11–28 (2009)
25. Prokopenko, M.: Guided Self-Organization: Inception, vol. 9. Springer Science & Business Media, Heidelberg (2013). https://doi.org/10.1007/978-3-642-53734-9
26. Shalizi, C., Crutchfield, J.: Computational mechanics: pattern and prediction, structure and simplicity. J. Stat. Phys. **104**(3), 817–879 (2001)
27. Shannon, C.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(1, 2), 379–423, 623–656 (1948)
28. Sperati, V., Trianni, V., Nolfi, S.: Evolving coordinated group behaviours through maximisation of mean mutual information. Swarm Intell. **2**(2), 73–95 (2008)
29. Trianni, V.: Evolutionary Swarm Robotics: Evolving Self-Organising Behaviours in Groups of Autonomous Robots, vol. 108. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-77612-3
30. Utro, F., Di Benedetto, V., Corona, D., Giancarlo, R.: The intrinsic combinatorial organization and information theoretic content of a sequence are correlated to the DNA encoded nucleosome organization of eukaryotic genomes. Bioinformatics **32**(6), 835–842 (2015)
31. Villani, M., Roli, A., Filisetti, A., Fiorucci, M., Poli, I., Serra, R.: The search for candidate relevant subsets of variables in complex systems. Artif. Life **21**(4), 412–431 (2015)