On the Effects of the Robot Configuration on Evolving Coordinated Motion Behaviors

István Fehérvári*, Vito Trianni†,§, and Wilfried Elmenreich*,‡

*Networked and Embedded Systems/Lakeside Labs, Alpen-Adria Universität Klagenfurt, Austria
Email: {istvan.fehervari, wilfried.elmenreich}@aau.at

†ISTC-CNR, Rome, Italy
Email: vito.trianni@istc.cnr.it

‡Complex Systems Engineering, Universität Passau, Germany

§IRIDIA-ULB, Brussels, Belgium

***Abstract*—The design of robotic controllers through evolutionary methods requires making a large number of choices about the experimental setup, which are often left to the expertise or naïveté of the experimenter. Although much attention is normally given to the fitness function or the genotype-to-phenotype mapping determining the robot controller, the robot configuration is often selected with little care. Yet, an ill-defined configuration—in terms of the selected subset of the sensory-motor system, or in the pre-processing of the raw sensor data—may be decisive in determining the failure of the evolutionary process. In this paper, we study the effect of different robot configurations on the ability to evolve efficient behaviors for a swarm robotics system. In this domain, the choice of a good configuration is fundamental as even small details can lead to large differences in the group behavior. To demonstrate the importance of the robot configuration, we test different alternatives and measure the group performance on a bi-objective scale. We find that different configurations not only have a strong effect on performance, but they also correspond to behaviors with radically different features concerning the organization of the group.**

I. Introduction

Evolutionary Robotics (ER) can be a powerful method for the automatic synthesis of robotic systems, as demonstrated in the research carried out in the last two decades [1], [2], [3], [4], [5]. However, despite the potential advantages of such an automatic methodology, the success/failure of an ER experiment is often left to the expertise/naïveté of the experimenter. Indeed, it is often the case that many different choices of the experimental setup are arbitrarily performed, without relying on a well-assessed methodology. The robustness and flexibility of the evolutionary method sometimes counterbalances ill-conceived setups, but this cannot be *a priori* guaranteed. The situation is possibly worsened for collective and swarm robotics, due to the greater variability and dynamism characterizing these systems. Trianni and Nolfi [6] recognized the need for an engineering methodology in ER, pointing to four different aspects that must be taken into account when designing an experiment: the sensory-motor system of the robots, the genotype-to-phenotype mapping, the fitness function and the ecology. They analyzed the challenges posed by the application of ER methods to swarm robotics and showed that a small difference in the communication protocol exploited by the robots had a huge impact on the scalability of the evolved solutions [6].

In this paper, we follow this line of thought by addressing an issue that is often disregarded in ER research, that is, the choice of the sensory-motor system of the robots, or, more generally, the *robot configuration*. The robot configuration subsumes the set of available sensors and actuators that are used in the ER experiment, together with the necessary pre-processing of raw sensor readings and the post-processing of actuator commands. In the collective/swarm robotics case, the robot configuration also includes the communication devices and protocols to allow the exchange of information between robots. In short, the robot configuration corresponds to everything that is at the *interface* between the control system of a robot and the robot's environment.

The robot configuration is typically not modified by an evolutionary process.[1] It is usually chosen through intuition or experience, often relying on the smallest set of sensors and actuators, minimizing pre- and post-processing in order to limit the intervention of the experimenter and then letting the evolutionary machinery optimize the system within the given constraints. We believe that this approach is ill-posed for evolving complex robotic systems, such as swarm robotics ones. Indeed, these systems are very sensitive even to minor changes in their configuration, which might easily lead to unpredictable or unwanted system behavior [6]. Therefore, even if the selection of the robot configuration may seem a trivial issue, a proper choice can hardly be done without *a priori* information on the effects it has on the evolutionary dynamics. In this paper, we corroborate this claim with a well-grounded experimental setup, and we point to the need of methodological tools that support the choice of the best robot configuration.

To demonstrate our claims, we have chosen to evolve a coordinated motion behavior ("flocking") for a swarm of autonomous wheeled robots. This is a common benchmark for

[1] But see [7] for a prototypical case of brain-body coevolution.

swarm robotics, and many different studies have dealt with it to demonstrate its feasibility in different contexts [8], [9], [10]. This is mainly due to the large body of knowledge available on both the biological systems displaying coordinated motion [11], [12], [13] and the underlying self-organizing process, which has been extensively discussed and understood [14], [15], [16], [17]. On this bases, we can design our evolutionary experiment to provide the robots with the relevant information needed to display flocking behavior.

Generally speaking, flocking can be obtained by three basic individual rules: collision avoidance, flock centering and velocity matching. The first two rules serve to achieve *cohesion* in the group, while the third one leads to the alignment of the individuals in a coherent direction, which is necessary for group *motion*. Collision avoidance and flock centering can be easily executed relying on the *distance* and the *bearing* of close neighbors. Instead, velocity matching usually requires the knowledge of their *relative heading*. In order to evolve flocking in a group of robots, we exploit group cohesion and motion as the objectives to be maximized by our evolutionary algorithm. We test different robot configurations, which are designed to provide distance, bearing and relative heading information. Finally, we compare the features of the evolved behaviors for their capability to display efficient coordinated motion in the group.

In this study, we evolve robotic controllers exploiting a multi-objective evolutionary algorithm [18]. Multi-objective evolution leads to a wide exploration of the search space at the cost of a more complex analysis of the solutions that constitute the obtained Pareto front. In contrast to optimizing for a single aggregated measure, a multi-objective approach explores all the possible trade-offs between possibly conflicting objectives such as group cohesion and motion, and therefore the evolutionary process can produce a more diverse set of candidate solutions. By investigating the evolved behavior at different points on the Pareto front, we can explore the effect of a given robot configuration at large, without constraints from an averaging function.

The paper is structured as follows: Section II presents a step-by-step guide on the setup of our evolutionary experiments and the choice of the robot configurations to test. A first comparison between the different configurations based on performance only is presented and discussed in Section III, and a deeper analysis of the evolved behavior belonging to the Pareto fronts is provided in Section IV. The paper is concluded in Section V with some final remarks.

II. Evolving Flocking Behavior

In this section, we describe the design choices made for the evolution of coordinated motion behaviors in a group of robots. Evolutionary experiments are performed in simulation using ARGoS, a simulator tailored for swarm robotics, which provides high-speed and accurate simulations [19]. The simulated robots model the *marXbot* platform [20]. These robots are equipped with a belt of evenly distributed RGB LEDs that allow signaling with different colors, and that can be perceived by the onboard omni-directional camera. We exploit the marXbot LEDs to define several robot configurations that can provide the required information for coordinated motion, as described in Section II-A. We complete the description about the evolutionary setup presenting the full sensory-motor configuration of the robots (Section II-B), the controller and the genotype-to-phenotype mapping (Section II-C), and the evolutionary algorithm along with the fitness function used (Section II-D).

A. LED configuration

According to the models of self-organized flocking [14], the relevant information needed by an individual to decide its action consists of the distance, the bearing and the heading of close neighbors. To convey such information we use the RGB LEDs around the robots, which allow to display various colored patterns. In our experiments, the chosen pattern will be always displayed to allow neighboring robots to detect each other, and possibly to obtain all the information required for moving in a coordinated way. To investigate the influence of a robot configuration on the evolved behavior, we test different LED configurations. In total there are 12 LEDs all around the robot circular body, each configurable to display any color in an 8-bit RGB spectrum (see the figure in Table I). We decided to restrict the number of colors to red and blue, because each color corresponds to additional inputs to the robot controllers, as detailed below. In the whole, for each of the 12 LEDs, we have 3 possible states—red (`R`), blue (`B`) and off (`0`)—that correspond to 3^{12} possible configurations. We selected a subset of configurations that intuitively convey information about the robot heading, which is crucial for effective coordinated motion. We defined both left-right and front-rear colored patterns, and we vary the number of LEDs used. Additionally, we also run two control experiments with "naive" LED configurations: the one with all LEDs off, and the other with all LEDs in the same state (`B`), therefore not conveying any heading information. As shown in Table I, in total we have 10 different configurations that can be grouped in three categories: the naive (1-2), the left-right (3-6) and the front-rear (7-10). The first configuration is only meant for testing if adding LEDs truly have an effect on performance, while the second one should determine if adding heading information is beneficial. The other two categories include setups with different number of LEDs turned on (respectively 1, 2, 4 and 6 LEDs). For the sake of simplicity, we considered only symmetric configurations.

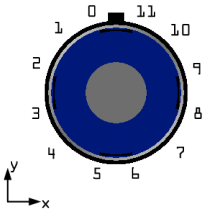
B. Sensory-motor System

Each robot is equipped with a minimal set of sensors and actuators, which is considered sufficient for displaying a flocking behavior. The selected LEDs in the robot configuration are always on, and can be perceived by the omnidirectional camera up to the distance of 1 meter. The image is processed to extract a list of $i = 1, \ldots, N$ red/blue color blobs $c \in \{r, b\}$ with their distance $\rho_{c,i}$ and angle $\theta_{c,i}$, resulting in a vector $\mathbf{v}_{c,i}(\rho_{c,i}, \theta_{c,i})$ for each detected blob (in polar coordinates).

TABLE I

WE TESTED 10 DIFFERENT LED CONFIGURATIONS, EACH DEFINED BY A CHARACTER ARRAY IN WHICH THE i^{th} ELEMENT REFERS TO THE STATE OF LED i, POSITIONS ARE AS INDICATED IN THE FIGURE ON THE RIGHT.

No.	Configuration	Description
1	000000000000	All LEDs are turned off
2	BBBBBBBBBBBB	All LEDs are turned blue
3	00B00000R000	
4	00BB0000RR00	1-6 LEDs indicating left
5	0BBBB00RRRR0	and right side
6	BBBBBBRRRRRR	
7	R00000B00000	
8	R0000BB0000R	1-6 LEDs indicating front
9	RR00BBBB00RR	and rear side
10	RRRBBBBBBRRR	



Additionally, each robot is equipped with 24 infra-red proximity sensors evenly distributed along the circumference of the robot's body. The typical range of these sensors is around 4-5 cm. Each sensor i provides a scalar value P_i inversely proportional to the distance of the object. For each sensor, we build a 2-dimensional vector $\mathbf{v}_{p,i}(\rho_i, \theta_i)$ in polar coordinates, where $\rho_i = P_i$ and θ_i corresponds to the sensor bearing. Measurement uncertainty is modeled by uniform noise within 5% of the input range.

Directly feeding the values obtained by the sensors would be straightforward, but would also present a huge search space for our evolutionary approach (24 proximity values + N colored blob values). Thus, some form of input preprocessing is necessary. To this purpose, we compute a single resultant vector for red blobs, for blue blobs and for the proximity sensors:

$$\mathbf{V}_k = \sum_i \mathbf{v}_{k,i}, \qquad k = r, b, p. \tag{1}$$

Then, we rescale the vector length to be within the range $[0, 1]$ by exploiting a sigmoid normalization:

$$\hat{\mathbf{V}}_k = \frac{\mathbf{V}_k}{|\mathbf{V}_k|} \frac{2}{1 + e^{-\beta|\mathbf{V}_k|}} - 1, \tag{2}$$

where $\beta = 2$ is a normalization parameter. Finally, we consider the projection along $M = 6$ equally distributed axes, by computing the scalar product:

$$I_{k,m} = \hat{\mathbf{V}}_k \cdot \mathbf{V}_m, \qquad m = 1, \ldots, M, \tag{3}$$

where $\mathbf{V}_m$ is the versor in the direction $(2m - 1)\pi/M$. In this way, we reduce the total number of scalar inputs to a more manageable size without significant loss of information. These values are then going to be fed to the robot's control software.[2] On the actuators side, given that LEDs are always kept in their state, the controller only commands the motors of the left and right wheels, which can linearly vary in the range $[-\omega_M, \omega_M]$, where ω_M is the maximum angular speed of the wheels ($\omega_M \approx 4.5\text{s}^{-1}$).

[2]Note that the heading information encoded in the LEDs color pattern (if any) is implicit. The available information is only given by the color vectors, and no assumption is made on how the neural controller should make use of it.

C. Genotype-to-phenotype mapping

All robots are completely identical both in body and control software (we use a homogeneous group). Therefore, we map the genotype to one single control structure that is cloned and instantiated separately for each robot. We employ a fully-connected feed-forward neural network without hidden layers. The neural network has 18 sensory inputs and 2 motor outputs. At each step the sensory neurons act as simple relays, while the output of the motor neurons is calculated as follows:

$$O_j = \sigma\left(\sum_i w_{ij} I_i + \beta_j\right), \quad \sigma(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

where I_i is the activation of the i^{th} input unit, β_j is the bias term, O_j is the activation of the j^{th} output unit, w_{ij} is the weight of the connection between the input neuron i and the output neuron j, and $\sigma(z)$ is the sigmoid function. The first 6 input neurons receive the input from proximity sensors: $I_m = I_{p,m}$. Neurons in the range $[7, 12]$ and $[13, 18]$ receive the data corresponding to the red and blue vectors: $I_{m+6} = I_{r,m}$ and $I_{m+12} = I_{b,m}$. Finally, the output of the two motor neurons is scaled onto the range $[-\omega_M, +\omega_M]$ and used to control the speed of the wheels. The bias terms and the connection weights of the network are genetically encoded parameters. Therefore, we have a direct encoding, meaning there exists a bijective function that relates the genotype to the phenotype.

D. Evolutionary algorithm and fitness function

In our experiments, we used a simple multi-objective evolutionary algorithm that operates on a population of 100 randomly generated genotypes. Each genotype contains the parameters of the control software of each robot as a vector of floating-point genes varying in the range $[-5, 5]$. After the evaluation of the performance of each individual of the population, the new population is created using a combination of elitism and mutation. In particular, the population is ranked according to the hypervolume metric [18] and the best 25 individuals are selected for reproduction: all individuals are retained unchanged in the next generation, while the rest of the population is generated by applying a mutation operator to copies of the elite individuals. Mutation is applied by adding to each gene a random value drawn from a normal distribution N(0,1), and trimming the value to keep it within the range $[-5, 5]$. The algorithm runs for a total of 200 generations.

Due to the fact that random initial conditions have an effect on the immediate performance of a candidate, each genotype is evaluated in 10 trials and the average performance over these trials is used to assess its fitness values. Each trial lasts $T = 120$ seconds corresponding to 1200 simulation steps. Initially, all robots are placed randomly within a circle with a diameter of 2 meters. Robots are rewarded for displaying coordinated motion, that is, they have to move as far as possible from the initial position while maintaining group cohesion. As a consequence, we defined a bi-objective function based on the following criteria: *cohesion* and *motion*. Cohesion

is maximized when the average distance between the robots and the geometric center of the group is minimized:

$$C = \max\left(0, 1 - \frac{1}{N}\sum_i \frac{|\mathbf{X}_i(T) - \hat{\mathbf{X}}(T)|}{d_m}\right), \quad (5)$$

where N is the number of robots, $\mathbf{X}_i$ is the position of robot i, $\hat{\mathbf{X}}$ the one of the group center of mass, and d_m is a normalization factor. Motion is computed as the total distance covered by the geometric center of the group:

$$M = \frac{|\hat{\mathbf{X}}(T) - \hat{\mathbf{X}}(0)|}{D_m(T)} \quad (6)$$

where $D_m(T)$ is the maximum distance a single robot can travel in T seconds.

III. Evaluating LED configuration

We performed 20 independent evolutionary runs for each configuration, each starting with different randomly generated populations. At the end of each evolutionary run, a post-evaluation procedure is employed where all candidates in the last generation are evaluated 300 times. To ensure a fair comparison between the different configurations, we use the same set of random seeds to initialize the evolutionary runs and to perform the post-evaluation. We exploit the Pareto-optimality relations to compare the results obtained with different configurations. In the simplest case, when one experimental condition is always dominated by another one (according to the $\lhd$-relation [21]), we can say that the latter gives a better approximation to the Pareto-optimal set. If such clear advantage cannot be determined, we can make use of attainment functions to extract information on the quality of different configurations [22]. The attainment function is associated to a given experimental condition, and it indicates the probability of a given point attaining (i.e., dominating or being equal) in the objective space. It thus characterizes statistically the output of a given experimental condition. Since this function is practically unknown, we approximate it using the simulation results (in particular, the obtained Pareto fronts of the 20 evolutionary runs we performed for each setup) obtaining the empirical attainment function (EAF) [23]. Once obtained the EAF for each condition, we can compute the difference between conditions to compare the relative quality of their output (see Figure 1 for an example). On this basis, we can comparatively analyze the effects of different robot configurations.

A. Naive configurations

The "naive" control conditions have been designed to make sure that robots can actually learn to profit from the extra information obtained by the use of LEDs and camera. Indeed, we could not rule out *a priori* the possibility that coordinated motion could be effectively performed exploiting the IR sensors only, or without heading information. Therefore, we initially test the `000000000000` and the `BBBBBBBBBBBB`

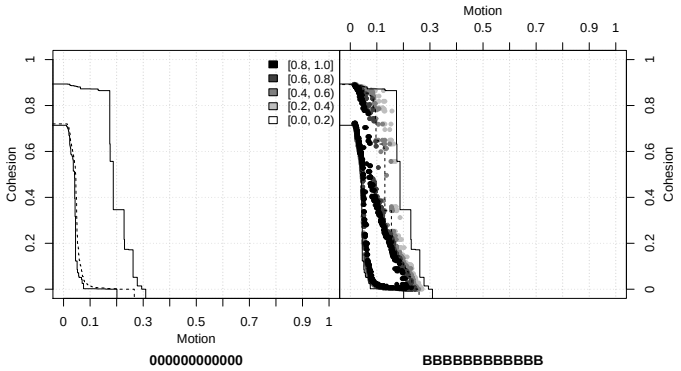


Fig. 1. Comparing naive configurations through empirical attainment functions (EAF). Left-side and right-side indicate the differential advantage of the `000000000000` and the `BBBBBBBBBBBB` configurations, respectively. Gray levels indicate the magnitude of the difference between the two setups: the darker the color, the larger the difference. For instance, a black point on the right indicates that the right-side condition attains that point in at least 80% more runs. Solid lines indicate the best and the worst surfaces, while the dashed line indicates the median. In this case, we observe a large advantage for the `BBBBBBBBBBBB` configuration.

configurations, which provide similar information about distance and bearing of close neighbors, the latter featuring a longer range thanks to the visual information from the camera.

As expected, the comparison of the obtained results for the two conditions shows that robots exploiting long-range visual information significantly perform better (see Figure 1), by systematically attaining higher values in both motion and cohesion. After investigating the evolved behaviors, we observe that no true coordinated motion has been achieved, as also suggested by the low motion performance of the attained points in the objective space. The increased performance with respect to the `000000000000` configuration can be definitely credited to the increased sensing range of the camera although it only helps the group to maintain coherence, which is not easily feasible on the basis of IR information only. We therefore obtain a confirmation that a suitable LED configuration is necessary for coordinated motion, and also that it must provide some information about the heading of the robots in order to obtain a good performance.

B. Left-right configurations

Left-right configurations are probably the most natural way of indicating the direction of a moving object, as this type of signaling is quite frequent in technical systems (e.g., visual signs of boats). However, choosing the right configuration for efficient coordinated motion with the marXbot robots remains a question. To find out how many LEDs should be turned on, and in what position, we ran tests with 1, 2, 4 and 6 LEDs, with different color on each side of the robot. Figure 2 shows a comparison of the tested configurations.

First, we compare the results of the uniform configuration (`BBBBBBBBBBBB`) with a left-right configuration exploiting the same number of LEDs (`BBBBBBRRRRRR`). The heading information provided by the latter can be actually exploited for better coordinated motion, as the difference between the

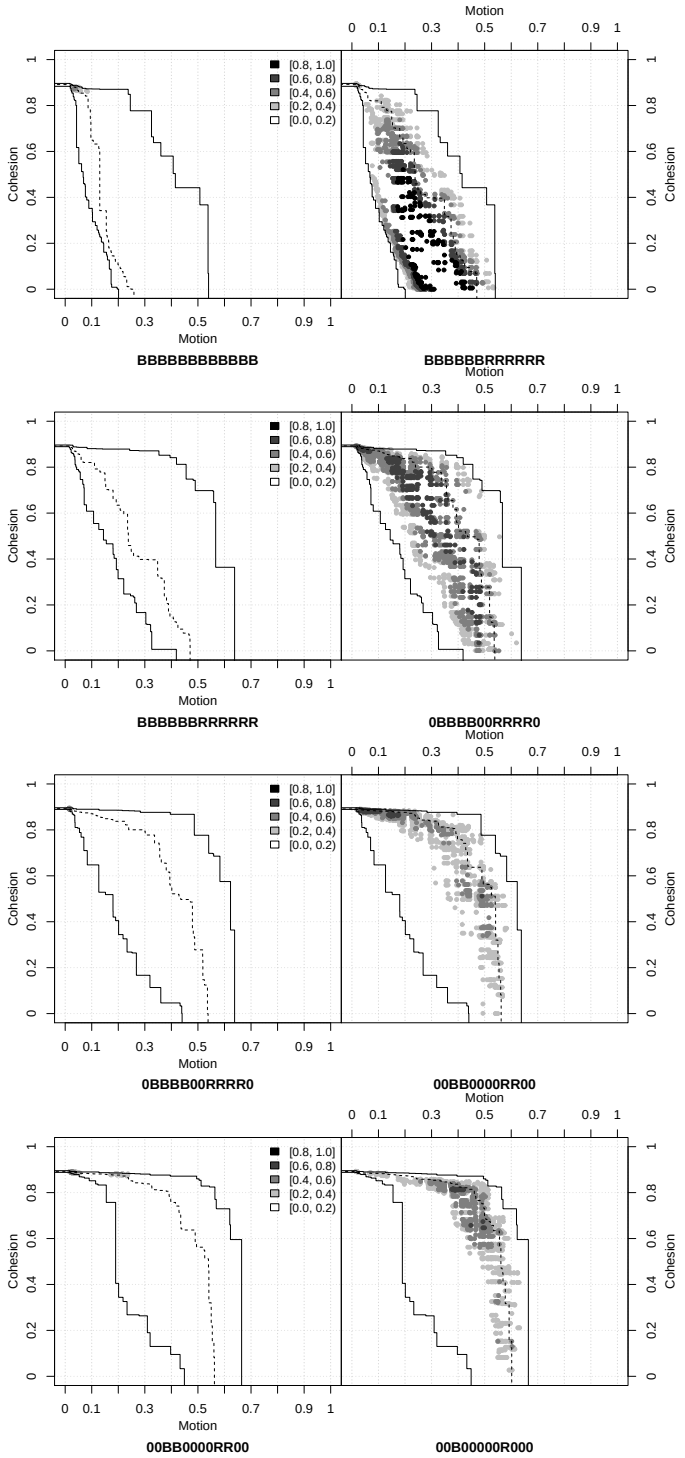


Fig. 2. Comparison among left-right configurations. The top graph compares the uniform configuration (BBBBBBBBBBBB) with a left-right configuration featuring an equivalent number of LEDs on (BBBBBBRRRRRR). The other graphs compare left-right configurations with decreasing number of LEDs. Setups with less LEDs perform better (see text for details).

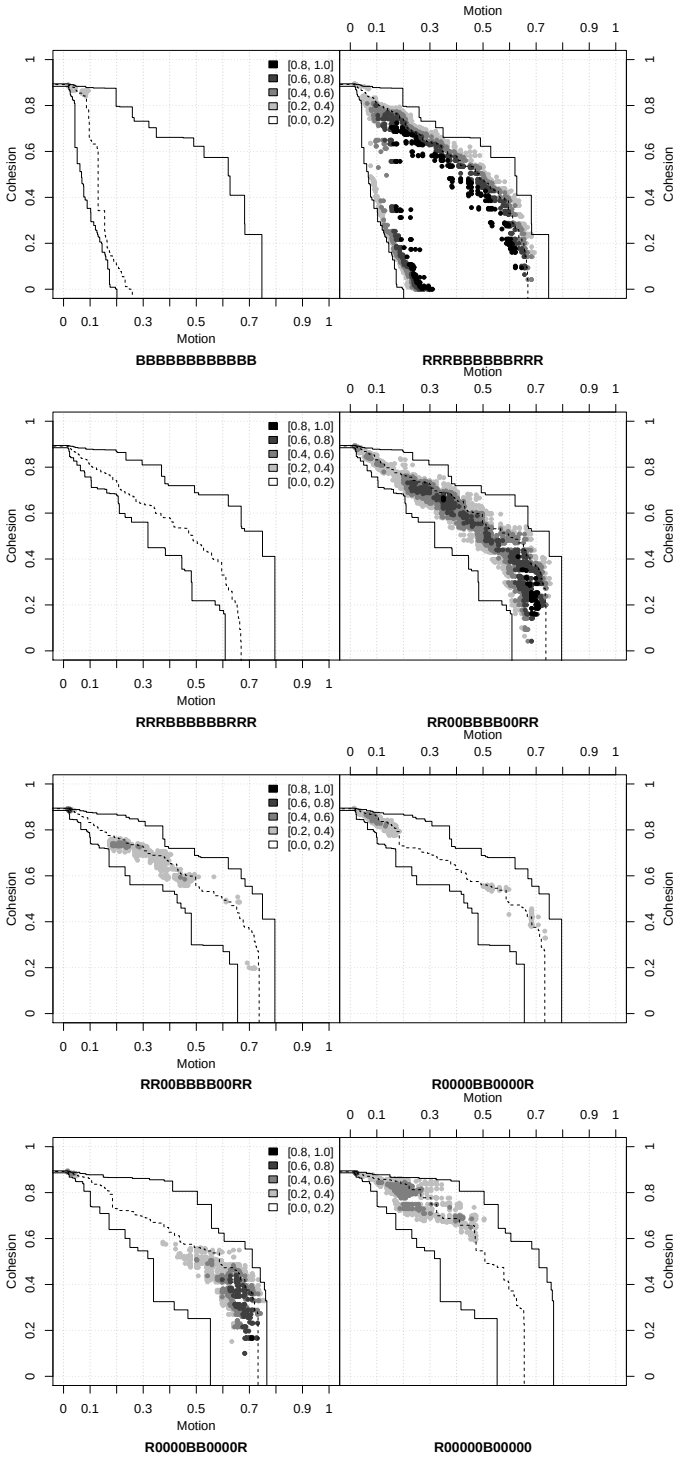


Fig. 3. Comparison among front-rear configurations (see text for details).

EAFs indicates (see top graph in Figure 2). If we consider cohesion only, the two setups are equivalent, but the left-right configuration definitely outperforms the uniform one as soon as motion is taken into account.

The comparison between different left-right configurations is performed for decreasing number of LEDs. We find that there is a inverse relationship between the number of LEDs used and the performance (see the second, third and fourth graph in Figure 2). Plus, this advantage appears at every part of the objective space, excluding only a small portion in which solely cohesion is maximized. We also note that the lower the

number of LEDs, the more the difference fades. A possible explanation for the advantage of less LEDs lies in our encoding of the visual data. The encoding presented in Section II-B might give less distinguishable information when two detected robots are very close. This is a counter-intuitive result as one would expect that the more information is available through LED signals, the better the quality of the coordinated motion behavior. Instead, minimal configurations seem to provide a selective advantage over the entire objective space. Apart from this, the results demonstrate that there is a strong influence of the configuration on the performance of the system.

C. Front-rear configurations

It is also possible to deduct the heading information from front-rear markers, as it is done for cars. Indeed, evolved strategies may exploit front-rear markers to obtain a coherent orientation of the group, which is the basics for coordinated motion. Therefore, we performed the same simulations as presented above with different number of LEDs used to signal the front and the rear of a robot. The comparison among the different configurations can be seen in Figure 3.

Also in this case, the comparison with the uniform configuration shows that front-rear markers can be actually exploited for coordinated motion. When comparing among different front-rear configurations, we note that it is not possible to draw a strong order of performance among them. While it is clear that the setup with 6 LEDs is strictly dominated by configurations with less LEDs, we note no clear difference between configuration with 4 and 2 LEDs and a differential advantage in separate zones of the objective space between configuration with 2 and 1 LEDs. Therefore, differently from the left-right configurations, front-rear markers appears to be exploited in a similar way despite their number.

D. Comparison between different configuration categories

We finally propose a comparison between left-right and front-rear configurations, performed among setups featuring the same number of used LEDs (e.g., `00BB0000RR00` vs. `R0000BB0000R`). In the case with 6 LEDs, the front-rear configuration demonstrates better performance in covering the objective space (see the top graph in Figure 4). As we decrease the number of used LEDs, the situation changes in favor of the left-right configuration, which starts to dominate in the top part of the objective space, where more and more solutions featuring high cohesion and good motion are found. As shown by the various graphs in Figure 4, the front-rear configurations keep an advantage in the areas of the objective space characterized by high motion but low cohesion. Thus, without further analysis it is impossible to determine the best configuration.

IV. Classification of the obtained solutions

Despite the usage of relatively simple configurations, we were unable to conclusively determine the best setup. Indeed, the average values of motion and cohesion alone do not indicate what kind of behavior the group is displaying. For this reason, it is necessary to observe the evolved behaviors and identify their characteristics in relationship to the area occupied in the objective space. By thorough analysis of the evolved behaviors on the entire Pareto front, we observed the following prototypes of group motion[3]:

Stationary: robots stay close to each other, showing minimal or no group motion.

Disperse: robots spread around with almost no cohesion.

Wavefront: robots move together forming a single-width arc.

Train: robots align and follow each other one by one like wagons of a train.

Flocking: robots show a healthy mix of cohesion and motion by displaying true flocking.

To objectively classify each solution into one of the above listed behaviors we defined a set of suitable metrics and a classification procedure. Besides motion and cohesion measures, we compute the following metrics at the end of each trial.

- The number of connected components K in a graph where each node corresponds to a position of a robot and an edge exists if and only if the distance between the robots is less than the maximum visual range. This measure serves to understand if the group splits in subgroups moving in different directions.
- The angle Θ between the average direction of motion of the group and the main axis of the group given by the position of the two robots that are farthest away from each other. This allows to distinguish between *trains* and *wavefronts*.

We performed 50 trials for every non-dominated solution of each configuration to obtain a reliable classification, and on the basis of this sample, we ran the classification procedure shown in algorithm 1. We first classify as "disperse" those behaviors in which the group splits into two or more smaller groups more than 25% of the time. This is measured as $Q3(K) > 1$, where $Q3(K)$ is the third quartile of the number

if $Q3(K) > 1$ **then**
  return *Disperse*
end
if $Q_3(M) \leq D_a$ **then**
  return *Aggregation*
end
if $Q_1(C) > D_c$ **then**
  return *Flocking*
end
if $Q_2(\Theta) \leq \frac{\pi}{4}$ **then**
  return *Train*
end
return *Wavefront*

Algorithm 1: Procedure for classification of different behaviors (see text for details).

[3]A video of the observed behaviors is available at http://www.youtube.com/watch?v=DMTajtJasTs

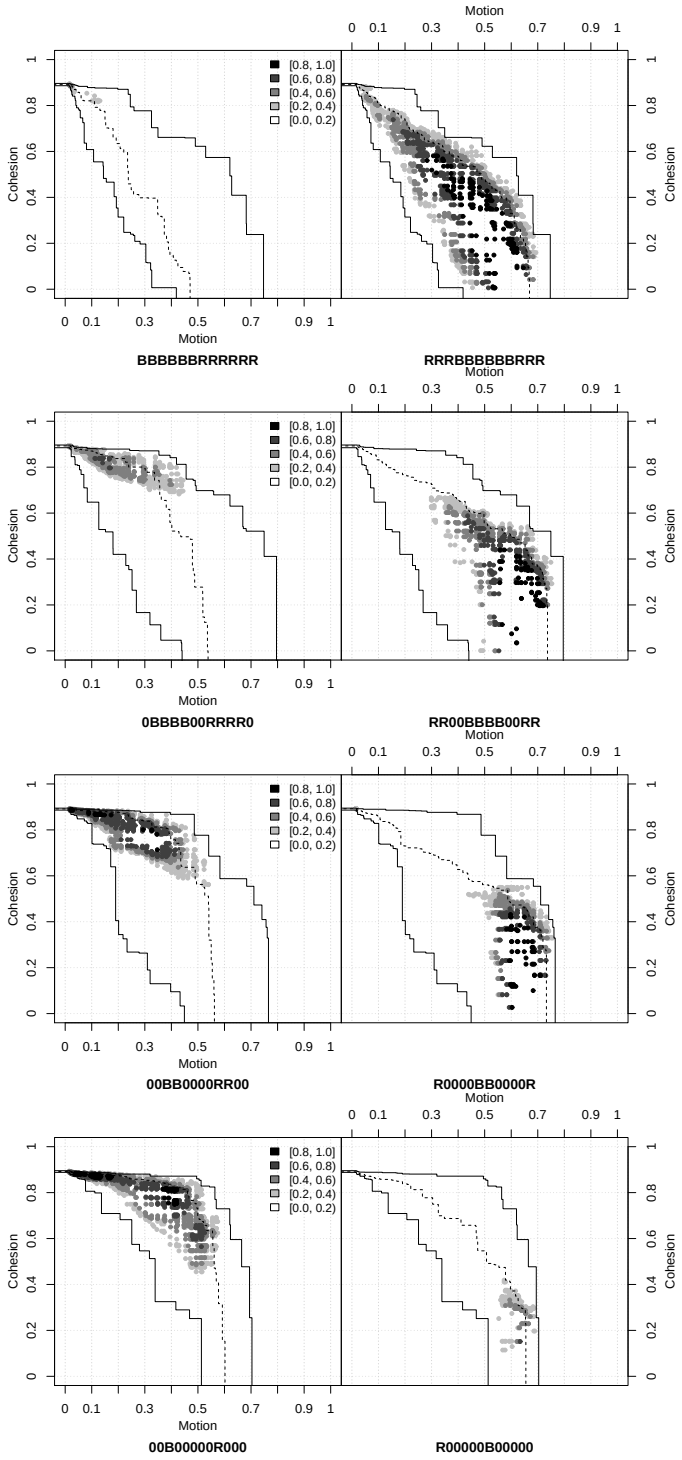


Fig. 4. Comparison of the left-right and front-rear configurations (see text for details).

of connected components.[4] Then, we identify the solutions belonging to the class "stationary", which exhibit very low motion values. We classify as "stationary" those behaviors that have $Q_3(M) \leq D_a$, where $Q_3(M)$ is the first quartile of the motion samples and $D_a = 0.25$ is an empirically selected threshold below which no significant motion of the group is observable. By using the third quartile we encompass in the class those behaviors that aggregate at least in 75% of the trials. At this point, we have removed all cases that do not present a good coordinated motion behavior. We then classify as "flocking" those behaviors that do not present an elongated shape, determined as $Q_1(C) > D_c$, where $Q1(C)$ is the first quartile of the cohesion samples and $D_c = 0.8$ is an empirically determined threshold. This class encompasses behaviors in which the group presents high values of cohesion in at least 75% of the trials. Finally, we distinguish between "train" and "wavefront" on the basis of the angle Θ, which indicates whether the main axis of the group is close to the average direction of motion of the group (train) or orthogonal to it (wavefront). We therefore look at the median value $Q_2(\Theta)$ and classify the behavior as "train" in case it is lower than $\pi/4$, and as "wavefront" otherwise.

Table II shows the behavior classification of every setup. We notice that left-right configurations do not produce "trains", and conversely front-rear configurations do not produce "wavefronts". Therefore, the LEDs configuration strongly determines the quality of behaviors that can be evolved. Moreover, we also notice that left-right configurations present an higher percentage of "flocking" behaviors, the lower the number of LEDs, the higher the probability to observe flocking. Therefore, the `00B00000R000` configuration seems to be the best one if flocking is the desired outcome.

Finally, we mapped the classification of behaviors over the corresponding points in the objective space, and we aggregate all solutions respectively for left-right and front-rear configurations (see Figure 5). It is possible to observe that different classes are rather separated on the objective space, with flocking behaviors occupying the area with high cohesion and variable motion. We also observe that "train" and "wavefronts" are specialized solutions, and that they are less clustered in the objective space, indicating a wide range of possibilities of displaying these behaviors in the different configurations. If we compare Figure 5 with Figure 4, we can understand how the ability to produce more flocking behaviors gives a performance advantage to the left-right configuration, while the front-rear configurations dominate mainly thanks to behavior in the "train" or "disperse" categories, which correspond to larger motion values. Again, from this analysis the left-right configuration results in better performance.

V. Conclusions and Future Work

We have shown the importance of the robot configuration on the ability to evolve efficient coordinated motion behaviors in a swarm of robots. Our results indicate that the selection of the robot configuration can determine the success or the failure of the evolutionary experiment. This raises the need of well-assessed methodologies to guide the experimenter in the choice of the robot configuration. In this respect, we envisage

[4]The rest of the classification is performed discarding those trials in which the group splits (which are anyway a minority), in order to avoid incorrect classifications.

TABLE II
DETAILED CLASSIFICATION RESULTS FOR EACH CONFIGURATION.

Configuration		Stationary		Disperse		Train		Wavefront		Flocking	
		#	%	#	%	#	%	#	%	#	%
3	00B00000R000	128	47,8%	45	16,8%	0	0,0%	1	0,4%	94	35,1%
4	00BB0000RR00	135	50,4%	50	18,7%	0	0,0%	4	1,5%	64	23,9%
5	0BBBB00RRRR0	155	57,8%	70	26,1%	0	0,0%	11	4,1%	41	15,3%
6	BBBBBBRRRRRR	211	78.7%	116	43,3%	0	0,0%	5	1,9%	11	4,1%
7	R00000B00000	200	74,6%	60	22,4%	41	15,3%	0	0,0%	32	11,9%
8	R0000BB0000R	180	67,2%	44	16,4%	90	33,6%	0	0,0%	20	7,5%
9	RR00BBBB00RR	168	62,7%	50	18,7%	122	45,5%	0	0,0%	15	5,6%
10	RRRBBBBBBRRR	185	69,0%	85	31,7%	82	30,6%	0	0,0%	8	3,0%

in future work the usage of automated methodologies. For instance, the robot configuration could be put under evolutionary pressure (i.e., as an additional objective in a multi-objective setup). However, suitable encoding must be devised to ensure the co-evolvability of configuration and behavior.

ACKNOWLEDGMENTS

This work was supported by the Austrian Science Fund (FFG) grant FFG 2305537 and Lakeside Labs via funding from the European Regional Development Fund and the Carinthian Economic Promotion Fund (KWF) under grants KWF 20214—21532—32604, and KWF 20214—22935—34445. Vito Trianni acknowledges support from the European Science Foundation project H^{2}Swarm partially funded by the Italian CNR and the Belgian F.R.S.-FNRS. We thank Lizzie Dawes for proofreading an earlier version of the paper.

REFERENCES

[1] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press/Bradford Books, Cambridge, MA, 2000.

[2] D. Floreano and L. Keller, "Evolution of adaptive behaviour in robots by means of darwinian selection," *Plos Biology*, vol. 8, no. 1, p. e1000292, 2010.

[3] R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*. MIT Press/Bradford Books, Cambridge, MA, 2007.

[4] I. Harvey, E. Di Paolo, R. Wood, M. Quinn, and E. Tuci, "Evolutionary robotics: A new scientific tool for studying cognition," *Artificial Life*, vol. 11, no. 1-2, pp. 79–98, 2013.

[5] I. Fehervari and W. Elmenreich, "Evolving neural network controllers for a team of self-organizing robots," *Journal of Robotics*, 2010.

[6] V. Trianni and S. Nolfi, "Engineering the evolution of self-organizing behaviors in swarm robotics: A case study," *Artif. Life*, vol. 17, no. 3, pp. 183–202, Aug. 2011.

[7] H. Lipson and J. Pollack, "Automatic design and manufacture of robotic lifeforms," *Nature*, vol. 406, no. 6799, pp. 974–978, 2000.

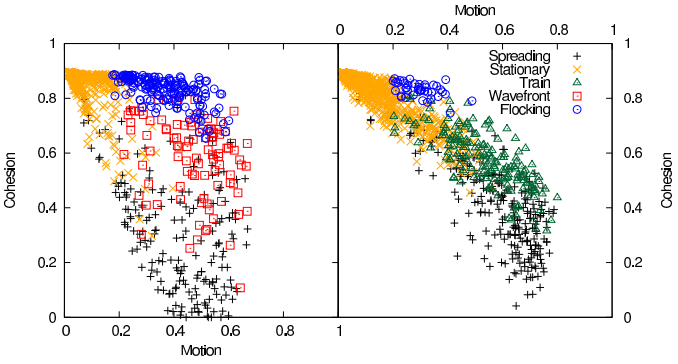


Fig. 5. Classification of left-right (left) and front-rear (right) configurations.

[8] A. Hayes and P. Dormiani-Tabatabaei, "Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02)*. IEEE, 2002, pp. 3900–3905 vol.4.

[9] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 97–120, 2008.

[10] S. Hauert, S. Leven, M. Varga, F. Ruini, A. Cangelosi, J. Zufferey, and D. Floreano, "Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 5015–5020.

[11] A. Huth and C. Wissel, "The simulation of the movement of fish schools," *Journal of Theoretical Biology*, vol. 156, no. 3, pp. 365–385, 1992.

[12] I. Couzin, J. Krause, R. James, G. Ruxton, and N. Franks, "Collective memory and spatial sorting in animal groups," *Journal of Theoretical Biology*, vol. 218, no. 1, pp. 1–11, 2002.

[13] A. Cavagna, A. Cimarelli, I. Giardina, G. Parisi, R. Santagati, F. Stefanini, and M. Viale, "Scale-free correlations in starling flocks," *Proceedings of the National Academy of Sciences*, vol. 107, no. 26, p. 11865, 2010.

[14] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '87. New York, NY, USA: ACM, 1987, pp. 25–34.

[15] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988–1001, June 2003.

[16] I. Fehérvári and W. Elmenreich, "Evolutionary methods in self-organizing system design," in *Proceedings of the 2009 International Conference on Genetic and Evolutionary Methods*, Las Vegas, NV, USA, 2009, pp. 10–15.

[17] T. Vicsek and A. Zafeiris, "Collective motion," *Physics Reports*, 2012.

[18] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[19] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[20] 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*, 2010, pp. 4187–4193.

[21] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, April 2003.

[22] V. G. da Fonseca, C. M. Fonseca, and A. O. Hall, "Inferential performance assessment of stochastic optimisers and the attainment function," in *Lecture Notes in Computer Science*. Springer, 2001, pp. 213–225.

[23] M. López-Ibáñez, L. Paquete, and T. Stützle, "Exploratory analysis of stochastic local search algorithms in biobjective optimization," in *Experimental Methods for the Analysis of Optimization Algorithms*, T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, Eds. Berlin, Germany: Springer, 2010, pp. 209–222.