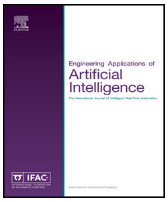
Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Research paper

Evolutionary swarm formation: From simulations to real world robots

Daniel H. Stolfi [a,*], Grégoire Danoy [a,b]

[a] Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 6, Avenue de la Fonte, Esch-sur-Alzette, L-4364, Luxembourg
[b] FSTM/DCS, University of Luxembourg, 6, Avenue de la Fonte, Esch-sur-Alzette, L-4364, Luxembourg

ARTICLE INFO

Keywords:
Swarm robotics
Evolutionary algorithm
E-Puck2
ARGoS simulator
Robot formation

ABSTRACT

Swarms of autonomous robots have become an interesting alternative for space and aerospace applications due to their versatility, robustness, and self-organising capability. Some of those applications, such as asteroid observation, convoy escort, and counter-drone systems, rely on stable formations achieved around a central point of interest. However, the use of different numbers of robots and the existence of a wide range of initial conditions contribute to make it a challenging problem. We propose in this research work a novel approach for self-organising a swarm of autonomous robots where the members' movements depend only on their relative position (range and bearing) obtained from their respective radio beacons. An optimisation approach based on an evolutionary algorithm is proposed to calculate the optimal swarm's parameters, e.g. speed and attracting/repelling forces, to achieve robust formations under different initial conditions and failure rates. Experiments are conducted using realistic simulations of six case studies featuring three, five, ten, fifteen, twenty, and thirty robots. The best valued configurations were tested on 420 scenarios showing that our proposal is robust since it has always achieved the desired circular formation. Finally, we have used real E-Puck2 robots to validate the swarm's capability of self-organising around a central point of interest as well as its resilience to robot failure, obtaining successful circular formations in all the experiments.

1. Introduction

Robot formation addresses the collaboration of a group of robots to follow a certain pattern while travelling in coordination through an area, maintaining a particular behaviour (Issa and Rashid, 2019; Cohen and Agmon, 2021). According to Beard et al. (2001), coordination between swarm members can be classified as leader-following (a global leader which navigates through the environment), behaviour-based (each robot independently decides its own behaviour), and virtual structure (centralised control via a global planner). On the other hand, Oh et al. (2015) propose the classification of formation control systems into three main approaches: position (using absolute global coordinates), displacement (using relative positions), and distance-based (using inter-agent distances).

Resulting robot formations are used for performing collective tasks, usually in an autonomous way. Proposed applications are surveillance (Brust et al., 2021), synchronisation of spacecrafts (Chung et al., 2009), salvage missions (Cardona and Calderon, 2019), caging and grasping (Makita and Wan, 2017), localisation and mapping (Saeedi et al., 2015), and representing dynamic deforming figures (Hauri et al., 2014). Several problems arise when working with formation of robots such as optimal initial locations and path planning to reach the final positions. Moreover, final positions which are defined in advance would restrict how the robots in formation can adapt themselves to variable real situations. This is especially true when the initial positions are unknown, sometimes even requiring the use of global coordinates on each robot or to be provided through a central node.

We propose an approach based on range and bearing, where the optimal distances between robots are automatically calculated to arrange a desired circular formation taking into account a point of interest at its centre. The coordination between robots follows a behaviour-based approach and the formation control system is distance-based. Hence, we address the aforementioned issues by not requiring any intelligent node (leader) nor an external system of coordinates (our proposal is distributed and decentralised). Our approach allows the robots to decide about their next movement direction only based on local information, being able to build successful formations even when some robots fail or there are communication issues. We have defined an exclusion zone around the formation centre to avoid being too close to the surrounded object while the formation algorithm itself avoids robot collisions. Additionally, full coverage is also taken into account in order to maximise the area of the object being observed by the swarm. It is worth mentioning that the robots self-arrange around a central object, encircling it, in contrast with other proposals where predefined arbitrary shapes are used.

[*] Corresponding author.
E-mail addresses: daniel.stolfi@uni.lu (D.H. Stolfi), gregoire.danoy@uni.lu (G. Danoy).

https://doi.org/10.1016/j.engappai.2023.107501
Received 15 June 2023; Received in revised form 19 October 2023; Accepted 11 November 2023

Possible applications derived from our proposal are asteroid observation, interferometry, convoy protection, and escorting a rogue drone to clear a no-fly zone. Other uses can also be possible, providing the swarm is large enough and the formation radius can be achieved without risking collisions. We have defined six case studies to test our proposal. They comprise swarms of three, five, ten, fifteen, twenty and thirty robots. We propose a set of configuration parameters for the system to be optimised by using an evolutionary algorithm to obtain stable formations in all the tested scenarios. Our main contributions are:

1. A 2D distributed formation approach for ground robots with the objective of surrounding a central object of interest.
2. An evolutionary algorithm specially designed and parameterised to efficiently calculate the optimal parameters of the formation system.
3. A simulation approach, including up to 30 robots, with the aim of studying failure tolerance, communication loss, and different detection ranges.
4. The validation of our proposal using up to ten real robots where the system resilience to robot failures was also tested.

This paper is divided into seven sections. The second section gives the works related to our current proposal. Our approach for robot formation system is in Section 3. Section 4 describes our optimisation algorithm and its parameterisation. Two alternative techniques are presented in Section 5 as competitors of our optimisation algorithm. Section 6 contains our experimental results and Section 7 brings conclusion and future work.

2. Related work

Some research works related to motion planning for autonomous robots as well as distributed formation control are reviewed in this section, see Liu and Bucknall (2018) and Dias et al. (2021) for two surveys.

A set of control schemes for nonholonomic mobile robots using a leader-following formation tracking is proposed by Liang et al. (2018). In this proposal the leader's speed is calculated using pixel coordinates and it is tested using simulations with five robots, and through experiments with two real robots. Other leader-follower approaches are proposed in Lin et al. (2021) and Kiełczewski et al. (2022). The former uses visual kinematics to achieve a steady-state response in the controller, while the latter uses localisation coordinates in order to mimic motion of the virtual leader. Both approaches were tested using the OptiTrack motion capture system. Our formation does not rely on a leader, all the robots run the same algorithm, and have the same role in the formation. It uses range and bearing to calculate the robots' relative positions and arrange them around a central object.

Deep Reinforcement Learning techniques have been applied to mobile robot navigation (Zhu and Zhang, 2021). Ma et al. (2020) study a control system for target encirclement using deep reinforcement learning and a deterministic policy gradient algorithm. The authors aim to overcome the difficulties of applying Q-learning to continuous action spaces. The experiments are conducted using the Gazebo simulator and four robots. Xie et al. (2021) use reinforcement learning to deal with the control of multiple unmanned surface vehicles in a leader-follower formation. The physical relationship between vehicles is defined and a deep deterministic policy gradient algorithm is proposed. The authors test their proposal using an *ad hoc* simulation platform with a V-shape formation and three, five and seven vehicles. Bezcioglu et al. (2021) optimise the parameters of a flocking mechanism using deep reinforcement learning to train a swarm of 100 robots. The authors use active elastic sheet as the dynamic model for the flocking mechanism which does not take into account orientation information. Sui et al. (2021) propose a novel method based on deep reinforcement learning to solve the formation control with collision avoidance problem. The training framework comprises two stages: imitation learning (controller) and reinforcement learning (reward). Experiments were conducted using simulations and an omnidirectional-wheeled car system. Jiang et al. (2019) designed a deep neural network to map the onboard LIDAR (Light Detection and Ranging) sensor in order to control the robot's motors using a decentralised formation control policy and a supervised learning framework. The experiments where performed using three robots and the VREP simulator. In Aldana-Franco et al. (2021) an architecture based on an artificial neural network is proposed to control one E-Puck robot performing two tasks: environmental cleaning and room interchanging. The network weights are optimised using a genetic algorithm and the experiments are conducted using the Webots simulator. Our approach is different form these research works as we use evolutionary optimisation for parameterising our formation problem, we experiment with up to thirty robots, and we validate our results by using the ARGoS simulator and real robots.

An algorithm which takes a given point cloud and transforms it into an acyclic directed graph was proposed by Li et al. (2018). The resulting graph is then used by the control law to allow a swarm of robots to progressively form the target shape is which each robot can find its position using two other robots. The robots know the graph representation is advance, although they are not assigned to a specific position. The proposed algorithm was verified using simulations and experiments on real Khepera IV robots. Our proposal is also tested using simulations and real robots although our formation algorithm is based on the relative positions of all the robots in the swarm and a central point of interest. In Li et al. (2019) the authors present a multi-robot team which keeps a given pattern taking into account the maximum energy consumed. A group of algorithms is proposed to control the swarm energy levels using a given number of charging stations and a behavioural state machine is in charge of coordinating the access to the different charging stations. After being tested on multiple simulations and also using physical robots (Khepera IV), this approach was capable of keeping their the energy levels into the desired limits. We are not evaluating energy consumption in our work and do not use predefined positions nor any hierarchy.

A decentralised multi-robot formation control is proposed by Lopez-Gonzalez et al. (2016) using robots equipped with distance and orientation sensors. A family of distance-based functions with collision avoidance is used to obtain control laws dependent on the data sensed and a communication graph. The final convergence of the formation was proven using Lyapunov stability theory. The authors present analytic results as well as experiments with three iRobotCreate robots. An index-free, pattern formation algorithm is presented by Queralta et al. (2019) to arrange robots in different shapes using individual spherical distributions. The authors improve the system robustness as the final robots' positions are interchangeable. They perform a numerical analysis and laboratory tests using Elektro-Monstertrucks and RPLiDAR technologies. These research works use local information from the robots' sensors as we do. However, they are focused on control theory and do not use optimisation techniques nor study failure tolerance. In our case we only define the distance to the central point of interest and the robots self-organise each other around it using the optimised swarm parameters.

Quantum computing has recently used for solving robotic swarm intelligence problems (Mannone et al., 2023). Gao et al. (2019) propose a new robot planning algorithm based on a quantum-inspired evolutionary algorithm. It works in a discretised environment and calculates an optimal robot planing path. Results from simulations indicate that the proposed algorithm can be used in static and dynamic scenarios. Chella et al. (2023) focus on decision-making and path planning collective tasks and present a quantum-based path planning algorithm for a swarm of robots. A logic gate implemented with a quantum circuit is used for modelling pairwise information-exchange,

Table 1
Comparison between the analysed related works and our formation proposal: Distributed Formation Algorithm (DFA).

Problem	Papers	Approach	Number of robots	
			Simulation	Real
Leader-Follower	Liang et al. (2018)	Leader-follower kinematics model	5	2
	Lin et al. (2021)	Leader-follower kinematics model	5	2
	Kiełczewski et al. (2022)	Artificial potential function	–	4
	Xie et al. (2021)	Deep deterministic policy gradient	7	–
	Sui et al. (2021)	Consensus theory	6	6
	Fazenda and Lima (2007)	Potential fields	4	–
	Mastellone et al. (2008)	Lyapunov-type approach	3	3
Swarm flocking	Bezcioglu et al. (2021)	Active elastic sheet	100	–
Environmental Tasks	Aldana-Franco et al. (2021)	Acetyl-modulated architecture	1	–
Self-reconfiguration	Mezghiche and Djedi (2020)	Real-observation quantum genetic algorithm	1	–
Shape	Jiang et al. (2019)	Decentralised control policies	3	–
	Li et al. (2018)	Directed acyclic graph	75	8
	Lopez-Gonzalez et al. (2016)	Distance-based potential functions	4	3
	Queralta et al. (2019)	Spherical indicator distribution	5	1
Path planing	Gao et al. (2019)	Quantum-inspired evolutionary algorithm	1	–
	Chella et al. (2023)	Logic gates & Grover-based path planning	10	–
	Ahmed et al. (2015)	Optimised potential fields	1	–
Target encircling	Ma et al. (2020)	Deterministic policy gradient	3	–
	DFA	**Attracting/Repelling forces**	**30**	**10**

while path planning uses the Grover's search algorithm, also implemented with a quantum circuit. Mezghiche and Djedi (2020) study the self-reconfiguration ability of a simulated modular robot. A quantum inspired genetic algorithm is proposed to optimise the weights of an artificial neural network with fixed architecture for the robot controller. Different robot configurations were tested for 100 time steps and ten simulation runs. Our approach uses an evolutionary algorithm to calculate the parameters of the swarm and it is tested in both, simulations and real world experiments.

Approaches based on potential fields (Khatib, 1990) combined with viscous friction (Louste and Liegeois, 2000) have been proposed as path planning methods for robots. Among them, some research works use a leader-follower approach (Fazenda and Lima, 2007), others optimise a PID controller using a Particle Swarm Optimisation (PSO) algorithm (Ahmed et al., 2015), or follow a trajectory avoiding collisions using local reference coordinates (Mastellone et al., 2008). The main difference with our proposal is that they use a global coordinate system (instead of only local information) to know the robot positions, potential fields are usually calculated in advance working as collision avoidance in a mapped scenario. Moreover, they do not use an evolutionary algorithm to tune the formation parameters, and do not propose a set of different scenarios to provide a variety of initial conditions.

A new discrete Mycorrhiza optimisation nature-inspired algorithm is presented in Carreon-Ortiz et al. (2022). This algorithm is inspired by the symbiosis between plant roots and a fungal network called the Mycorrhizal network. The authors perform a study comparing their proposal with other bio-inspired algorithms in terms of standard deviation.

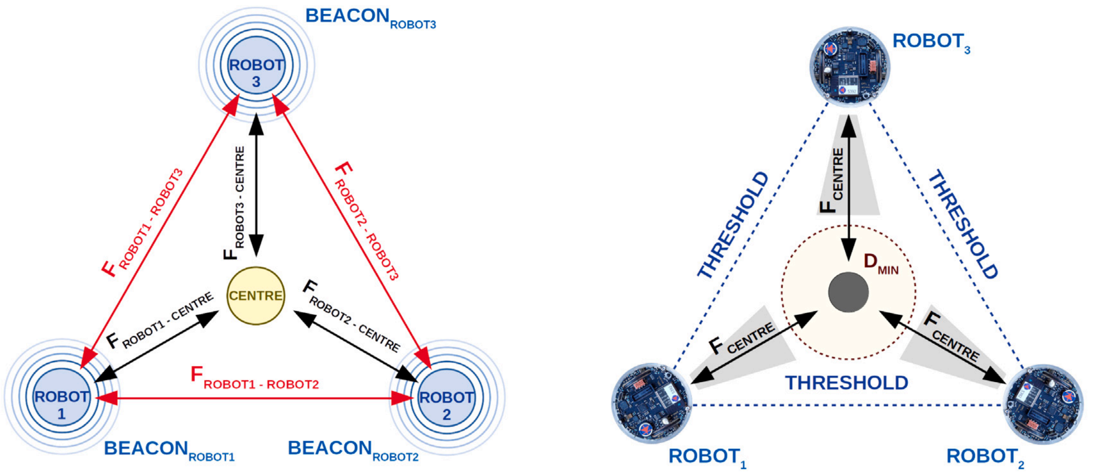
In general, all these works use techniques to achieve formations which are different from our proposal. In the present article we achieve robot formations through the use of forces that balance attraction and repulsion among robots. The forces are optimally tuned by a set parameters calculated by a metaheuristic, i.e. an evolutionary algorithm. External obstacles are not present, although collision between robots are avoided thanks to the same forces that contribute to shape the formation. We have used a simulator for robot swarms to implement and evaluate our formation algorithm on 600 scenarios consisting of swarms with different number of robots and diverse initial positions. Our study also includes real world tests using ground robots. Table 1 summarises the differences between the analysed related works and our proposal.

We have given some initial steps (Stolfi and Danoy, 2022) towards this new mature approach. In our present work we have identified and analysed new parameters which directly modify the behaviour of the formation algorithm. We have increased the number and complexity of the case studies to perform more exhaustive tests. We have proposed and parameterised a new hybrid evolutionary algorithm to optimise the aforementioned parameters and obtain more stable and resilient formations. Moreover, we have performed a comparison with other formation and optimisation techniques, addressed robot failures, and detection range restrictions, as well. To our knowledge, a research work addressing the optimisation of the parameters of a swarm of robots with the objective of achieving robust 2D formations, using an evolutionary algorithm plus its validation via the ARGoS simulator and actual E-Puck2 robots, has not been proposed before.

3. Robot swarm formation

Stable robot formation systems are challenging since they have to address different constraints consisting of absence of absolute coordinates, limited communication range including packet loss, and uncertain initial robot positions, just to name a few. We propose a swarm formation system in which autonomous robots arrange around a central object of interest in a formation that conserves a predefined desired distance to the centre (D_{centre}). A robot in the swarm receives only a beacon signal (solely carrying the robot's id) from the other robots which is used to calculate the relative orientation and distance to its counterparts. At no time do they know their absolute localisation such as GPS coordinates, nor the desired final positions in the formation, only local information would be used for each robot's decisions. Additionally, a detection mechanism, e.g. another radio signal in our study (although it can be images from a camera, LIDAR data, etc.), is required to identify and track the central object of interest. The formation is collaboratively built by the robots, which arrive from different directions, by equilibrating attracting and repelling forces, converging to a given polygon whose vertices depend on the number of robots. The system considers two types of forces according to their origin and destination: (i) between the robots in the swarm and (ii) between each robot and the formation centre. Fig. 1(a) presents the system diagram where three robots are depicted showing their beacons and the attracting/repelling forces involved in the formation.

When designing our formation system, we have observed how the robot behaviours affect the final formation in terms of shape and stability. Hence, we have proposed modifying these characteristics by setting up four variable parameters of the swarm: a $threshold$ to trigger an attracting/repelling action between robots, the minimum distance D_{min} to



(a) System diagram showing the attracting and repelling forces

(b) Real world model showing the E-Puck2 robots and the system's parameters

Fig. 1. System diagram and real world model showing the attracting/repelling forces involved in the formation as well as the proposed parameters.

avoid a central exclusion zone, the strength of the attracting/repelling force F_{centre}, both in relation to the centre, and the *speed* of the robots. We have experimentally tested all these four parameters concluding that they are all needed to achieve final stable formations. Fig. 1(b) shows a real world implementation of the formation model and its related parameters. Algorithm 1 describes our proposed Distributed Formation Algorithm (DFA). It controls the robots' movement, thus it is executed by each robot's CPU using the same parameterisation, including the desired distance to the centre D_{centre}.

Algorithm 1 Distributed Formation Algorithm (DFA).

```
function DFA($D_{centre}, threshold, D_{min}, F_{centre}$)
    $\vec{r} \leftarrow \vec{0}$
    for $robot \in BEACONS$ do
        $range, bearing \leftarrow Range\_and\_Bearing(robot)$
        $r_x \leftarrow r_x + (range - threshold) \times \cos(bearing)$
        $r_y \leftarrow r_y + (range - threshold) \times \sin(bearing)$
    end for
    $range, bearing \leftarrow Range\_and\_Bearing(centre)$
    $\omega \leftarrow 1.0$
    if $range < D_{min}$ then
        $\omega \leftarrow F_{centre}$
    end if
    $r_x \leftarrow r_x + \omega(range - D_{centre}) \times \cos(bearing)$
    $r_y \leftarrow r_y + \omega(range - D_{centre}) \times \sin(bearing)$
    $\Theta \leftarrow \arctan \frac{r_y}{r_x}$
    return $\Theta$
end function
```

The DFA begins initialising the vector r which will be used to calculate the influence of the rest of the robots in the swarm and the attracting/repelling force to/from the centre. Then, the range and bearing from the other robots are obtained using each beacon received, so that the components of the r can be calculated according to the given *threshold*. Next, the relative range and bearing of the central object is also obtained, so that depending of the distance to it and D_{min}, an extra repelling force F_{centre} can be used (ω) to avoid being to close to the centre of the formation. Finally, once the 2D components of r have been calculated, the next moving direction Θ is obtained and returned to the robot's controller.

The following sections describe the problem, analyse the parameter sensitivity, present the optimisation approach as well as our competitor algorithms, followed by our experiments and results. Fig. 2 shows a general schema of our experimental approach. First, a sensitivity analysis will be done to confirm the relevance of the DFA's parameters. Second, our Evolutionary Algorithm (EA) will be parameterised and used to optimise our case studies. Third, the best parameters will be used to test the DFA on different scenarios, including fault situations and restricted detection ranges. Finally, these parameters will be used in the real world validation to test the DFA on E-Puck2 robots.

3.1. Problem representation and evaluation function

The swarm formation problem is defined by the vector of parameters $\vec{x}$ shown in Eq. (1), where $threshold$, D_{min}, and F_{centre}, are the parameters of the DFA, and *speed* is the moving speed of the robots in the swarm. We have experimentally set the constraints for each parameter, so that $threshold$ and D_{min} are in the range $\left[\frac{1}{3} \times D_{centre}, 5 \times D_{centre}\right]$, F_{centre} is in the range $\left[\frac{10}{3} \times D_{centre}, 50 \times D_{centre}\right]$, and *speed* takes values between 0.1 and 2.5 (between 0.5 and 12.5 cm/s).

$$\vec{x} = \{threshold, D_{min}, F_{centre}, speed\} \tag{1}$$

The evaluation of each configuration requires the use of a simulator as there is no mathematical function to represent the complexity of this problem. The proposed fitness function $F(\vec{x})$ (Eq. (2)) calculates the values of ε and σ for a given configuration $\vec{x}$ through the simulation of M different initial positions (scenarios). It obtains a numerical value representing how far are the robots in the swarm to achieve a successful formation. The error ε (Eq. (3)) is calculated as the difference between the average distance μ from the robots to the centre of the formation and the desired distance D_{centre}. Eq. (4) shows the calculation of the average distance for the N robots in the swarm using the module of the distance vector $\vec{d_i}$. Finally, the standard deviation σ is calculated as shown in Eq. (5), where N is the number of robots, $\vec{d_i}$ is the distance of each robot to the centre, and μ is the previously calculated average distance of the robots to the centre. All the distances are in metres as they are collected from the simulator once the simulation has ended (final state of the formation).

Since we are simultaneously averaging the individual evaluations of M different scenarios, the achieved solution is expected to be more robust and general instead of being extremely fitted to just one initial set of positions. Our objective is to minimise the value of $F(\vec{x})$ (fitness) by using our optimisation algorithm to converge to configurations that

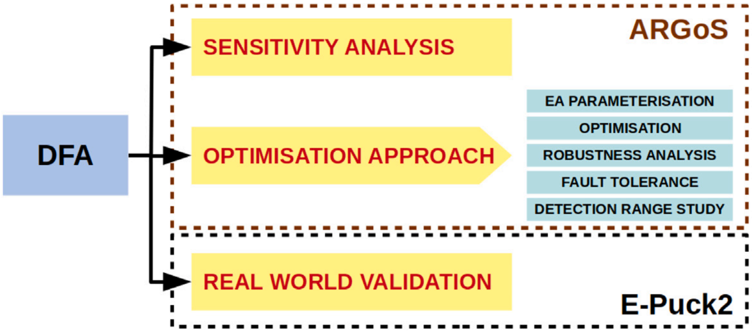


Fig. 2. Schema showing our experimental approach. The sensitivity analysis, the parameterisation of the EA and the optimisation of the DFA's parameters are performed using the ARGoS simulator. Several tests and studies are also performed in ARGoS, and finally, the best parameters found are validated using real E-Puck2 robots.

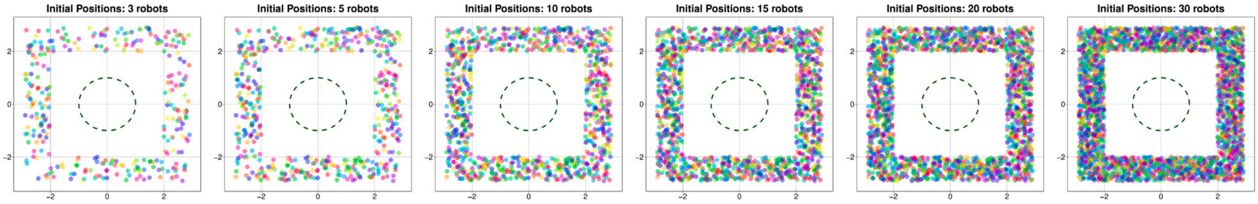


Fig. 3. Initial positions of robots for the 100 proposed scenarios of each case study. A circle showing the predefined formation radius is also depicted in a dashed line.

favour stable formations, placing the robots at the desired distance D_{centre} from the central object. Hence, the lower the fitness, the better.

$$F(\vec{x}) = \frac{1}{M}\sum_{j}^{M}\left[\varepsilon_j(\vec{x}) + \sigma_j(\vec{x})\right] \tag{2}$$

$$\varepsilon(\vec{x}) = \mu(\vec{x}) - D_{centre} \tag{3}$$

$$\mu(\vec{x}) = \frac{1}{N}\sum_{i}^{N}\|\vec{d}_i\| \tag{4}$$

$$\sigma(\vec{x}) = \sqrt{\frac{1}{N}\sum_{i}^{N}\left[\|\vec{d}_i\| - \mu(\vec{x})\right]^2} \tag{5}$$

3.2. Case studies

We have set up six case studies involving swarms of three, five, ten, fifteen, twenty, and thirty robots. We have calculated 100 different scenarios per case study in which the robots begin to build the formation from different initial positions in the studied area. We have followed this approach since in a real world situation the robots would usually arrive from unknown positions. Our intention is to avoid just visiting a few particular cases, addressing a more generic problem instead. Hence, we randomly selected 30 scenarios per case study to be used in the optimisation process. After obtaining one optimal configuration per case study (six in total) (Section 6.1), they would be tested on the remaining 70 unseen scenarios, as part of the robustness study, as explained later in Section 6.2.

The initial positions of the robots in the 100 scenarios are shown in Fig. 3 where it can be seen that a myriad of situations were covered, especially when the number of robots is high. We can see that no robot was initially placed close to the central area, as they are supposed to approach the central object from the borders, as expected in real world scenarios. These scenarios were modelled in ARGoS (Pinciroli et al., 2012), a simulator using multi-physics models able to efficiently simulate swarms of robots. The simulated robots were E-Puck2 (GCtronic, 2023) implemented as an ARGoS *plug-in* (Stolfi and Danoy, 2023) (Fig. 4). Each robot is aware only of the relative angle and distance to the other swarm members, both calculated using the received beacon signal. Although we have used an area of 4 × 4 metres, it can be scaled by modifying its parameters according to the new dimensions.

The controller of the E-Puck2 robots (both simulated and real) was programmed to move ahead continuously while the difference between the current moving direction and Θ is less than 15 degrees, as shown in Fig. 5. When a higher difference is provided by our DFA, the robot would stop moving to begin rotating accordingly. Once the angle difference is reduced and the robot reaches the desired orientation, it begins to move forward again. It has shown to be safer than using curved trajectories in order to avoid robot collisions. Although there is a repelling force between robots, if at any moment a moving robot (ADVANCE) detects another robot around it, an evasion action is taken consisting in moving away in the direction opposite to the detected robot(s). After that, the DFA will resume its execution once the surrounding space is cleared.

3.3. Analysis of the parameter sensitivity

We have performed an analysis of the parameters' sensitivity aiming to identify the level of importance of the parameters proposed for our swarm formation system. We used a method based on computing incremental ratios for each system's parameter as proposed in Morris (1991). This method first, selects a random sample of the parameters' values. Then, it uses the trajectories of measurement points in the parameters' space to analyse the distribution of values. Two sensitivity measures are computed: the mean μ (the overall influence of the parameter on the output) and standard deviation σ of the parameter distribution (the ensemble of the high order effects of the parameter). We have used the revised mean called $\mu\star$ to provide a more reliable ranking (Campolongo et al., 2007) as it solves the problem of the effects of opposite signs which occurs when the model is non-monotonic. Additionally, we have added a "fake" parameter, ω, which has no influence in the swarm's behaviour, in order to assess if the other proposed parameters are indeed necessary.

After the evaluation of 600 configurations of each case study, we obtained the pair of values (σ and $\mu\star$) for each parameter, depicted in Fig. 6. Each evaluation was performed using 30 selected instances of each case study (108,000 evaluations in total). We can observe that when the number of robots is low (three and five), the $threshold$ seems to be the most relevant parameter and that for twenty and thirty robots, D_{min} is apparently showing the biggest influence over the system, as the more robots, the more probable it is to enter the central zone. The

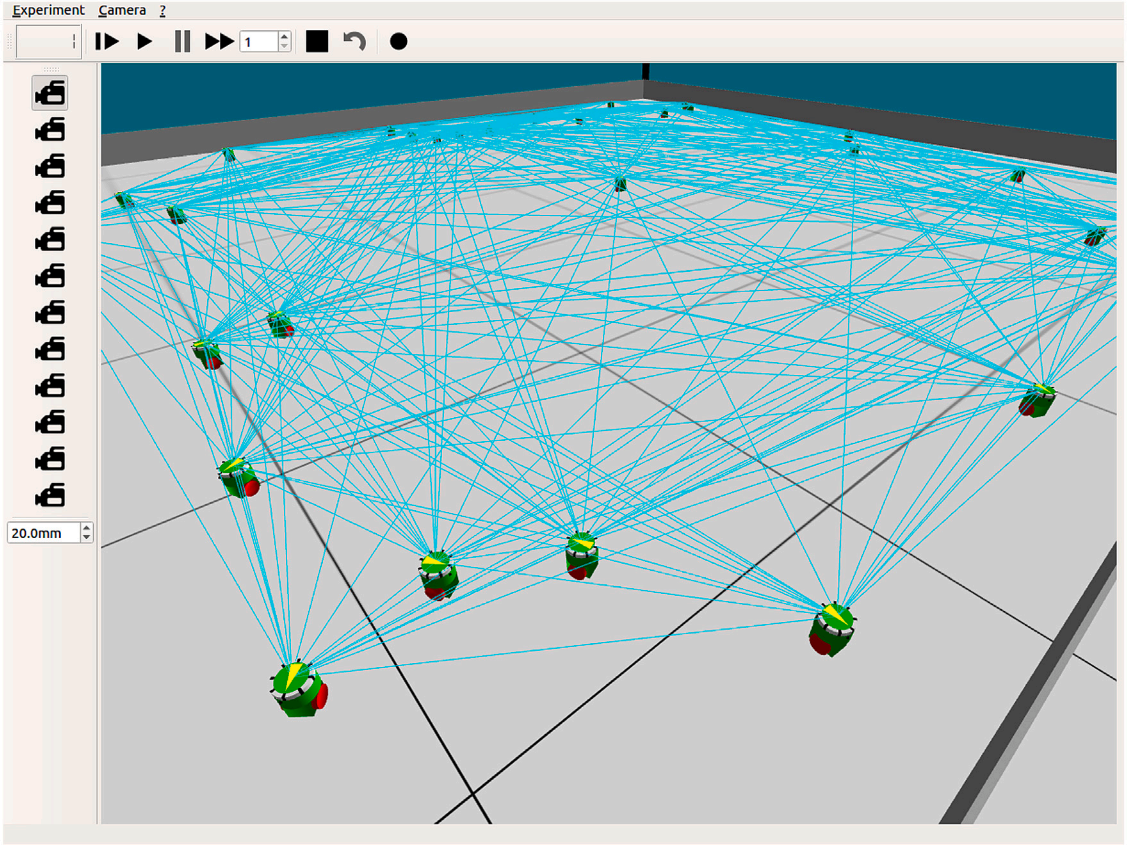


Fig. 4. Snapshot of the ARGoS simulator running one of the formation scenarios.

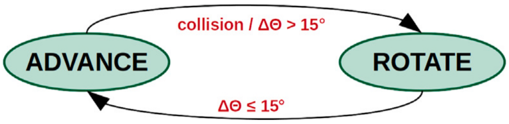


Fig. 5. States of the control system running on each robot.

case study featuring fifteen robots shows that the *threshold* and D_{min} have both an important influence on the resulting formation, while for ten robots all the system's parameters have a comparable effect on the formations achieved. It can be seen that in all the studied cases, ω (the "fake" parameter) has shown no influence on the system. Consequently, we see evidences that all the four proposed parameters do affect the swarm's behaviour while their importance depends on the amount of robots in the scenario.

4. Optimisation approach

The proposed swarm formation system requires to be optimally parameterised in order to achieve stable formations that do not depend on the initial positions of the robots (scenarios). Having that in mind, we have designed an Evolutionary Algorithm (EA) adapted to continuous optimisation, so that it calculates the optimal robots' parameters to obtain a polygonal shape, keeping the desired formation radius, while avoiding collisions. In the following sections we describe in detail our EA and its parameterisation using an automatic methodology. The use of an EA is not strictly linked to the DFA. Other optimisation techniques can be used instead, such as Particle Swarm Optimisation (PSO) (Kennedy, 2010) or Simulated Annealing (SA) (Kirkpatrick et al., 1983), providing they are able to efficiently manage the dimension of the solution space of this problem.

4.1. Evolutionary Algorithm (EA)

The proposed EA (Bäck and Schwefel, 1993) follows an evolutionary approach inspired by nature. It includes processes such as natural selection, reproduction, recombination and mutation of the individuals' genes, as well as the dominance and persistency of the best adapted individuals. Our proposed EA is divided into two sections or stages. The first one is a generational Genetic Algorithm (GA) having a population of μ individuals from which an offspring of λ individuals is obtained. Thus, the working population (Q) has 28 individuals in our study (the same amount as the population Pop). Our proposed GA temporarily stores each new generation of individuals in Q, which are stochastically exposed to crossover and mutation. After that, they would make their way to the population Pop only if they improve the existing ones (have a better fitness). The second stage applies a Local Search algorithm (LS) beginning by the best solution achieved by the GA. This High-level Relay Hybridisation (HRH) approach (Talbi, 2013) makes sense to ensure that the best found solution is in fact a local optima. We have applied LS after the GA (instead of doing so at each generation) to avoid premature convergence due to having very competitive solutions at the beginning of the algorithm. The pseudocode of our proposed EA can be seen in Algorithm 2.

Beginning at $t = 0$, $Q(0)$ is initialised, and $Pop(0)$ is filled with random individuals generated by the *Initial_Population* function. Next, the algorithm's main execution loop begins and continues until the 900 evaluations are performed (termination condition). Binary Tournament (Goldberg and Deb, 1991) was used as selection operator,

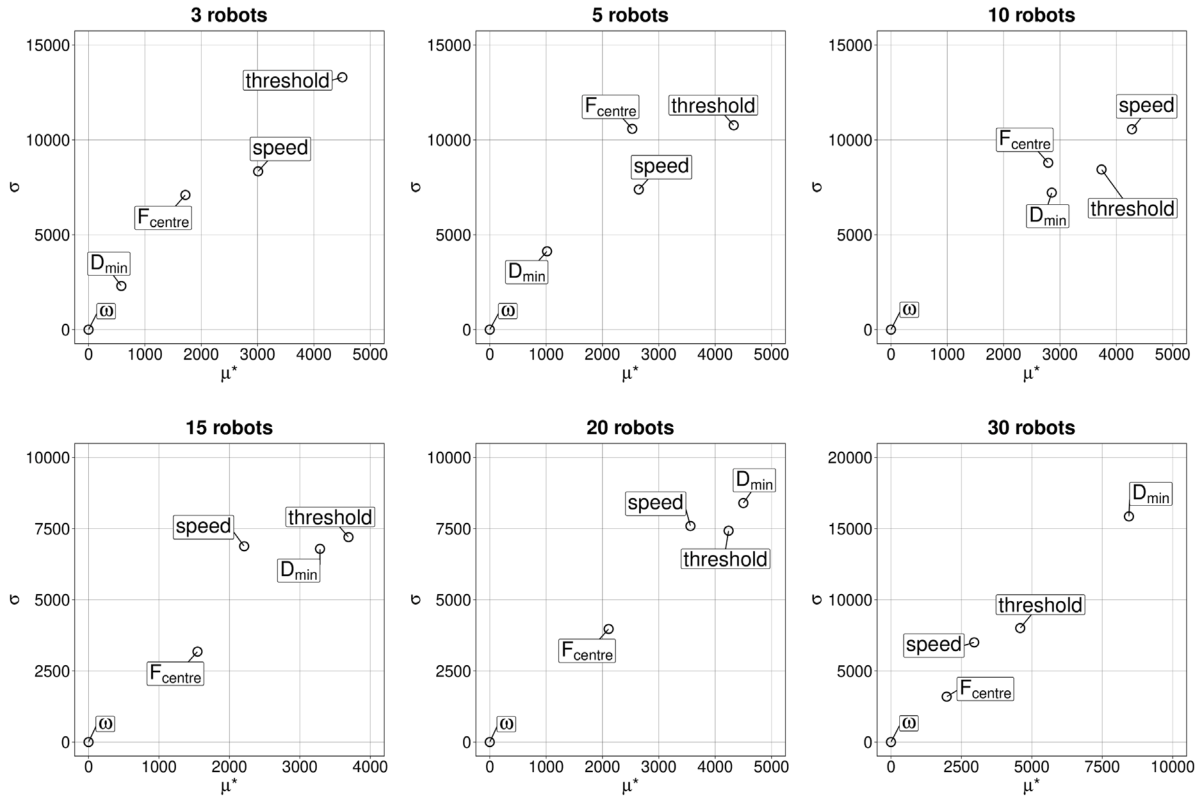


Fig. 6. Study of the parameter sensitivity of the proposed formation system (σ vs. $\mu\star$).

Algorithm 2 Evolutionary Algorithm (EA).

```
function EA($N_i$, $P_c$, $P_m$)
    $t \leftarrow 0$                                                   ▷ Genetic Algorithm
    $N_{ev} \leftarrow 0$
    $Q(0) \leftarrow \emptyset$                                        ▷ Q=auxiliary population
    $Pop(0) \leftarrow Initial\_Population(N_i)$                       ▷ Pop=population
    while $N_{ev}$ < MAX_EVALUATIONS × 0.9 do
        $Q(t) \leftarrow Selection(Pop(t))$
        $Q(t) \leftarrow Crossover(Q(t), P_c)$
        $Q(t) \leftarrow Mutation(Q(t), P_m)$
        $Evaluation(Q(t))$                                             ▷ Also increases $N_{ev}$
        $Pop(t+1) \leftarrow Replacement(Q(t), Pop(t))$
        $t \leftarrow t+1$
    end while
    $best(t) \leftarrow Best\_Solution(Pop(t))$
    while $N_{ev}$ < MAX_EVALUATIONS do                                ▷ Hill Climbing
        $Solutions(t) \leftarrow Neighbours(best(t))$
        $Evaluation(Solutions(t))$                                     ▷ Also increases $N_{ev}$
        $best(t+1) \leftarrow Best\_Solution(Solutions(t) \cup best(t))$
        $t \leftarrow t+1$
    end while
    return $best(t)$
end function
```

Table 2
Parameterisation of our EA calculated by using irace.

Robots	N_{ev}	μ	λ	p_c	p_m
3	1000	28	28	0.92	0.28
5	1000	28	28	0.55	0.51
10	1000	28	28	0.27	0.56
15	1000	28	28	0.87	0.40
20	1000	28	28	0.98	0.38
30	1000	28	28	0.99	0.61

Single Point Crossover (De Jong, 1975) was the chosen recombination operator, Integer Polynomial Mutation (Deb, 2001) was used for mutations, and elitism was used to update the population at the end of each generation (replacement). The parameters of the GA stage were calculated using the irace package, see Section 4.2. Once the GA stage ends, the best solution is taken to be used by the LS implemented by the Hill Climbing algorithm (Lin, 1965) (HC). It will explore the neighbourhood of the best solution for other 100 evaluations. We have used this hybrid approach to first explore the search space and later exploit the best optimal solution found.

4.2. Parameterisation of EA

The package irace (López-Ibáñez et al., 2016) was used to automatically calculate the EA parameters. Irace implements an iterated racing procedure. It uses Friedman test to perform a non-parametric analysis of variance by ranks (Derrac et al., 2011). Parameters of the EA's such as crossover probability (P_c) and mutation probability (P_m), were calculated by irace through 500 experiments, including all the case studies, where the elimination test has a confidence interval of 0.95. Table 2 shows the best parameter configurations calculated by irace and the remaining parameters of our EA. Additionally, the testing frequency of each parameter is shown in Fig. 7.

We can see that for three, fifteen, twenty, and thirty robots, the values obtained for crossover probabilities are according to the expected, i.e. $P_c > 0.8$, while for five and ten robots they are lower than 0.6. In most of the cases, the mutation rates are also higher than the commonly used $1/L = 0.25$, which we believe is due to the necessity of

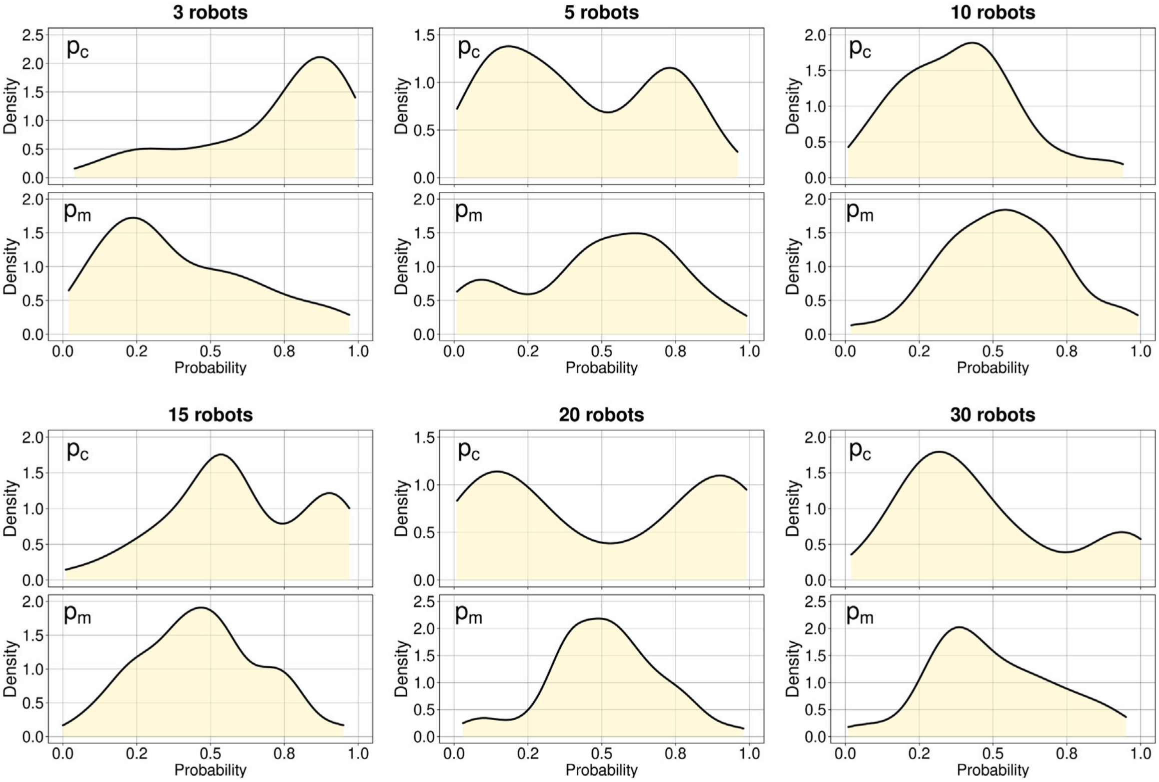


Fig. 7. Parameters sampling frequency per case study.

performing an intensive exploration of the search space. These results support the use of a systematic parameterisation method where the outcomes are sometimes different from what could have been guessed in advance. The remaining parameters, i.e. the amount of evaluations (N_{ev}), μ, and λ were set to match the architecture of the available experimentation nodes, ensuring a realistic optimisation time, since the evaluation of each configuration of our problem implies a time demanding simulation.

5. Algorithm competitors

Since we want to validate our EA as a reliable and precise method for configuring the formation system, we propose two alternative algorithms as competitors: A Geometric Approach and the system parameterisation using Random Search.

5.1. Geometric approach

First, we propose a Geometric approach in which we calculate the exact distance between adjacent robots (D_{robot}) in the polygonal formation (those linked by a virtual edge) and use it as the only parameter of the swarm as shown in Eqs. (6) and (7), where α is the angle between adjacent robots and N is the number or robots in the swarm. The resulting values for our proposed case studies ($D_{centre} = 1.0$) are shown in Table 3 where the distance D_{robot} decreases when the number of robots increases as it was to be expected.

$$D_{robot} = 2 \times \cos\frac{\alpha}{2} \times D_{centre} \tag{6}$$

$$\alpha = 180° \times \frac{N-2}{N}, \quad N \in \{3, 5, 10, 15, 20, 30\} \tag{7}$$

Table 3
Values for α and D_{robot} for the Geometric approach ($D_{centre} = 1.0$).

Robots	α	D_{robot}
3	60°	1.732
5	108°	1.176
10	144°	0.618
15	156°	0.416
20	162°	0.313
30	168°	0.209

5.2. Random Search (RS)

A sanity check for our EA is proposed by using a Random Search (RS) algorithm as an alternative way of calculating the optimal parameters for the formation system. The RS algorithm is presented in Algorithm 3. Initially, the first random solution is generated and selected as the best solution. After that, a new random solution is generated and compared with the current best one according to their fitness value. If the current solution is better (lower fitness value), the best solution will be replaced by the current one, or it is discarded otherwise. The same process is repeated until the maximum number of evaluations is reached, i.e. 1000 evaluations, as we wish a fair comparison between algorithms, with the evaluation by simulations being the most demanding operation.

6. Experiments and results

The conducted experiments are divided into six sections. First, the optimisation of each case study by using the proposed EA and RS is

Table 4
Results of the optimisation of 30 different scenarios per algorithm and case study ($D_{centre} = 1.0$).

Robots	Alg.	$threshold$ (best)	D_{min} (best)	F_{centre} (best)	$speed$ (best)	Fitness Min.	Fitness Median	Fitness Max.	Shapiro p-value	Wilcoxon p-value
3	RS	1.73	2.40	3.98	0.40	1.211	2.858	4.875	0.156	<0.001
	EA	1.73	1.29	4.27	0.40	**1.095**	**1.204**	**1.261**	0.042	
5	RS	1.62	1.81	3.90	0.50	1.461	4.708	8.552	0.906	<0.001
	EA	1.62	1.32	4.94	0.40	**1.104**	**1.162**	**1.593**	<0.001	
10	RS	1.58	1.49	4.80	0.40	1.342	5.545	12.878	0.333	<0.001
	EA	1.58	1.04	4.86	0.40	**1.311**	**1.526**	**3.348**	<0.001	
15	RS	1.57	3.76	2.30	0.40	2.575	8.149	26.380	<0.001	<0.001
	EA	1.58	1.15	4.52	0.40	**1.446**	**1.888**	**9.476**	<0.001	
20	RS	1.57	1.80	2.30	0.50	3.040	13.422	40.456	0.018	<0.001
	EA	1.57	1.21	2.80	0.40	**2.011**	**2.647**	**6.597**	<0.001	
30	RS	1.58	4.16	2.15	1.10	5.127	31.522	66.229	0.470	<0.001
	EA	1.57	1.95	2.24	0.40	**1.581**	**3.534**	**17.800**	<0.001	

Algorithm 3 Random Search (RS).

```
function RS
    $N_{ev} \leftarrow 0$
    $best\_solution \leftarrow Random\_Solution()$
    while $N_{ev}$ < MAX_EVALUATIONS do
        $temp\_solution \leftarrow Random\_Solution()$
        if $F(temp\_solution) < F(best\_solution)$ then
            $best\_solution \leftarrow temp\_solution$
        end if
        $N_{ev} \leftarrow N_{ev} + 1$
    end while
    return $best\_solution$
end function
```

presented in Section 6.1. Second, in Section 6.2, a robustness analysis is conducted, including also the results of the Geometric approach. The formation system is tested under several robot failures conditions in Section 6.3. Reduced detection ranges are tested in Section 6.4. A scalability study is done in Section 6.5. And finally, a real world validation is performed in Section 6.6 using E-Puck2 robots.

6.1. Optimisation results

The optimisation process consisted in performing 30 independent runs of EA and RS per case study. Thirty scenarios were optimised not only to get optimal formations but also to increase the system robustness. The optimisation algorithms were implemented using the jMetalPy package (Benítez-Hidalgo et al., 2019). The parallel runs were executed in nodes equipped with Intel Xeon Gold 6132 and a maximum of 128 GB of RAM, part of the HPC platform of the University of Luxembourg (Varrette et al., 2014). Realistic simulations always demand long execution times. The whole optimisation process (360 runs) took the equivalent to 244.9 h (approximately 10.2 days).

Table 4 shows the results of the optimisation process using EA and RS. In addition to the fitness values (minimum, maximum and median), the best configurations ($threshold$, D_{min}, $F_{(centre)}$, and $speed$) achieved by the algorithms are also reported. We can see that our EA has converged to better solutions (lower fitness values) than RS in all the case studies. The harder the problem the bigger the differences between the fitness values of EA and RS. The fitness median value of RS is more than twice the EA's for three robots, whereas for thirty robots the RS' results are nine times worse. The best values of $threshold$ and $speed$ calculated by the algorithms are similar in most cases, while D_{min} and F_{centre} present bigger differences.

Furthermore, we have tested the normality of the results using Shapiro–Wilk test showing that the majority of them are not normally distributed. Consequently, we have reported the median value and used non-parametric statistics, e.g. the Wilcoxon test, to compare the results. Indeed, the calculated Wilcoxon p-value shows that the differences between algorithms are statistical significant (p-value < 0.001). Moreover, the distribution of the optimisation results (30 runs per case study) are represented in Fig. 8 through boxplots. Despite the outliers, the better results achieved by the EA can be also deducted from the graphical representations. We can conclude that the use of an intelligent optimisation techniques such as our EA, is required to obtain optimal configurations of the proposed formation system. In the following section we test these best configurations found, on 70 unseen scenarios per case study to address their robustness.

6.2. Robustness analysis

This section presents the test of the best solutions obtained by the three analysed approaches in order to evaluate their robustness when used on the 70 unseen scenarios of each case study. Table 5 shows the results achieved in terms of real distances ($D_{robot}\star$ and $D_{centre}\star$), measured when the formations were stable (or after 1000 s if stability was not possible). Additionally, the percentage of successful scenarios and the elapsed times for these cases are reported, as well as the p-values from the Shapiro–Wilk and Wilcoxon tests. We can see that three robots present similar results for the three approaches, where the configurations calculated by RS and EA have achieved the same minimum, maximum, and median values for the measured $D_{centre}\star$. Results of the Geometric approach are also precise, having a $D_{centre}\star$ equal to 1.0 in almost all the scenarios with three robots. All the approaches have also achieved a desired stable formation in the 70 scenarios with three robots.

When we used five robots, the Geometric approach began to fail, being unable to place all the robots at the desired distance to the centre. This approach would also fail to achieve the desired formation in the rest of the case studies. RS' and EA's results were again approximately the same, achieving a 100% formation success rate. This is also true for ten robots although RS begins to show less precise values as the case studies grow in complexity. The most evident case is for thirty robots where RS has only achieved 88.6% of successful formations. In contrast, EA did achieve a stable formation at the desired distance from the centre along the 70 tested scenarios with thirty robots.

All the results were tested for normality showing Shapiro–Wilk p-values less than 0.003. Consequently, we have used the Wilcoxon test to compare our results as in the previous section. We can see that for three, five, fifteen, and thirty robots the reported results are statistically significant while for ten and twenty robots the results of EA and RS seem to be related, despite the fact that the best median and precision were always achieved by the EA.

Fig. 9 shows the evolution of the average distances (70 scenarios) from the robots to the central point ($D_{centre}\star$) and the distance between the robots in the swarm ($D_{robot}\star$), for each algorithm and

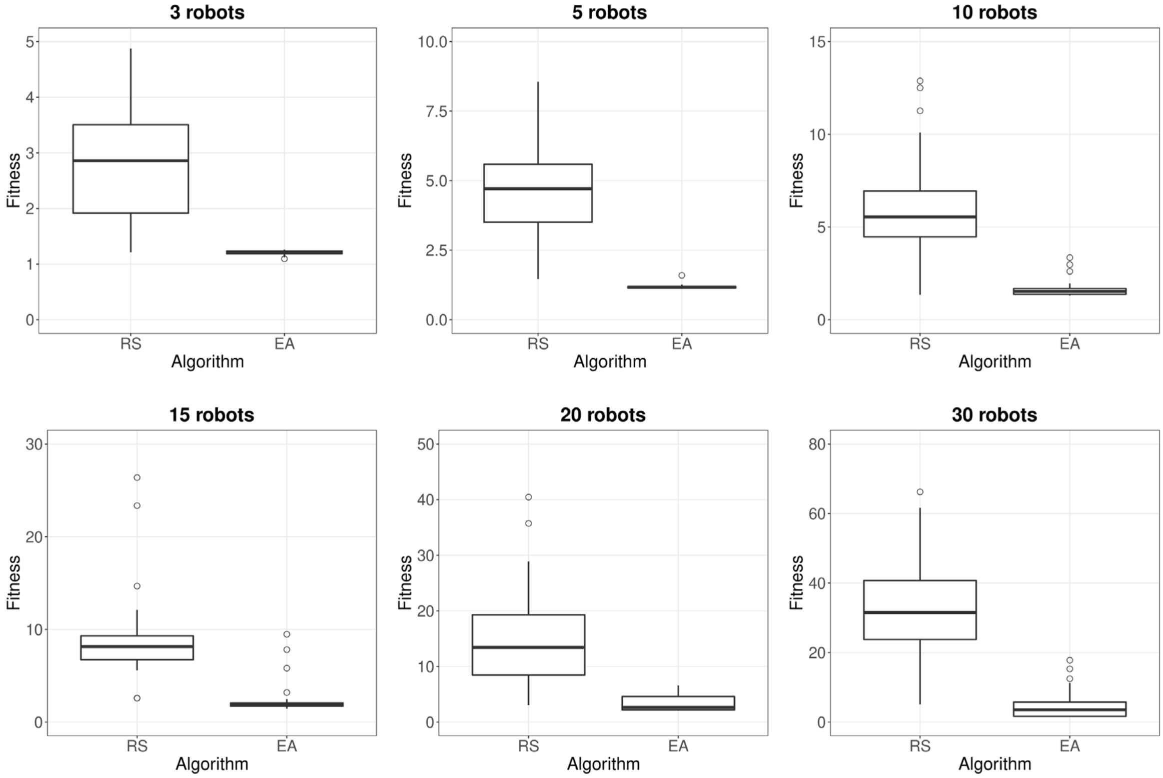


Fig. 8. Distribution of the results achieved after the optimisation of 30 training scenarios per case study. EA is not only more accurate than RS but also obtains better (lower) fitness values.

Table 5
Results obtained when testing the three different approaches on 70 scenarios ($D_{centre} = 1.0$). Distances are in metres and times are in minutes. The best results are in bold.

Robots	Approach	D_{robot}★			D_{centre}★			Formation	Elapsed time			Shapiro p-value	Wilcoxon p-value
		Min.	Median	Max.	Min.	Median	Max.		Min.	Median	Max.		
3	Geometric	1.726	1.730	1.735	0.994	0.999	1.004	**100.0%**	2.585	3.065	3.407	0.002	<0.001
	RS	1.726	1.731	1.733	**0.998**	**1.000**	**1.004**	**100.0%**	3.387	5.171	6.520	<0.001	0.003
	EA	1.724	1.732	1.734	**0.998**	**1.000**	**1.004**	**100.0%**	1.728	4.064	5.647	<0.001	–
5	Geometric	1.186	1.188	1.190	0.766	0.772	0.772	0.0%	–	–	–	<0.001	<0.001
	RS	1.536	1.538	1.540	0.997	**1.000**	**1.002**	**100.0%**	3.337	4.465	5.500	<0.001	0.025
	EA	1.537	1.538	1.540	**0.998**	**1.000**	**1.002**	**100.0%**	1.998	3.444	5.473	<0.001	–
10	Geometric	0.529	0.530	0.737	0.105	0.997	1.818	0.0%	–	–	–	<0.001	<0.001
	RS	1.390	1.402	1.403	0.913	**1.000**	1.059	**100.0%**	3.763	4.587	7.183	<0.001	0.210
	EA	1.402	1.402	1.404	**0.997**	**1.000**	**1.002**	**100.0%**	2.405	3.386	5.915	<0.001	–
15	Geometric	0.353	0.355	0.886	0.103	**1.000**	2.772	0.0%	–	–	–	<0.001	<0.001
	RS	1.355	1.357	1.434	0.846	0.998	2.267	98.6%	3.802	4.218	5.315	<0.001	<0.001
	EA	1.355	1.360	1.361	**0.919**	1.001	**1.067**	**100.0%**	2.268	3.205	6.865	<0.001	–
20	Geometric	0.316	0.389	1.349	0.100	0.507	3.532	0.0%	–	–	–	<0.001	<0.001
	RS	1.335	1.336	1.336	0.933	**0.999**	**1.048**	**100.0%**	3.448	3.941	4.468	<0.001	0.516
	EA	1.335	1.336	1.337	**0.936**	**0.999**	1.055	**100.0%**	2.182	3.380	5.960	<0.001	–
30	Geometric	0.410	0.577	1.695	0.101	0.569	3.084	0.0%	–	–	–	<0.001	<0.001
	RS	1.319	1.320	1.400	0.740	1.003	3.127	88.6%	2.673	2.828	3.255	<0.001	<0.001
	EA	1.315	1.315	1.315	**0.997**	**0.999**	**1.002**	**100.0%**	2.317	2.714	3.142	<0.001	–

case study. We can observe that five robots and more, using the Geometric approach, have trespassed the inner safe zone set to one third of the desired D_{centre} (green circles in Fig. 9), and that they have not converged to 1.0 when building the formation. RS' configurations presented different robot trajectories from EA, hence the different final values observed. Both approaches, RS and EA, have not caused collisions between robots nor trespassed the central safe zone showing that the DFA is performing well even when the parameters are not the optimal.

Fig. 10 shows the final positions of the robots in formation after testing the different approaches on each case study. Confirming the conclusions drawn from Table 5, the graphs show that more than three robots using the Geometric approach were unable to achieve the desired formation. The best configurations obtained by RS worked well except for fifteen and thirty robots where some scenarios were unsuccessful. Finally, the swarms configured by EA achieved satisfactory formations in all the 420 testing scenarios what confirms its competitive results already reported in Table 4 in the previous Section. Only the

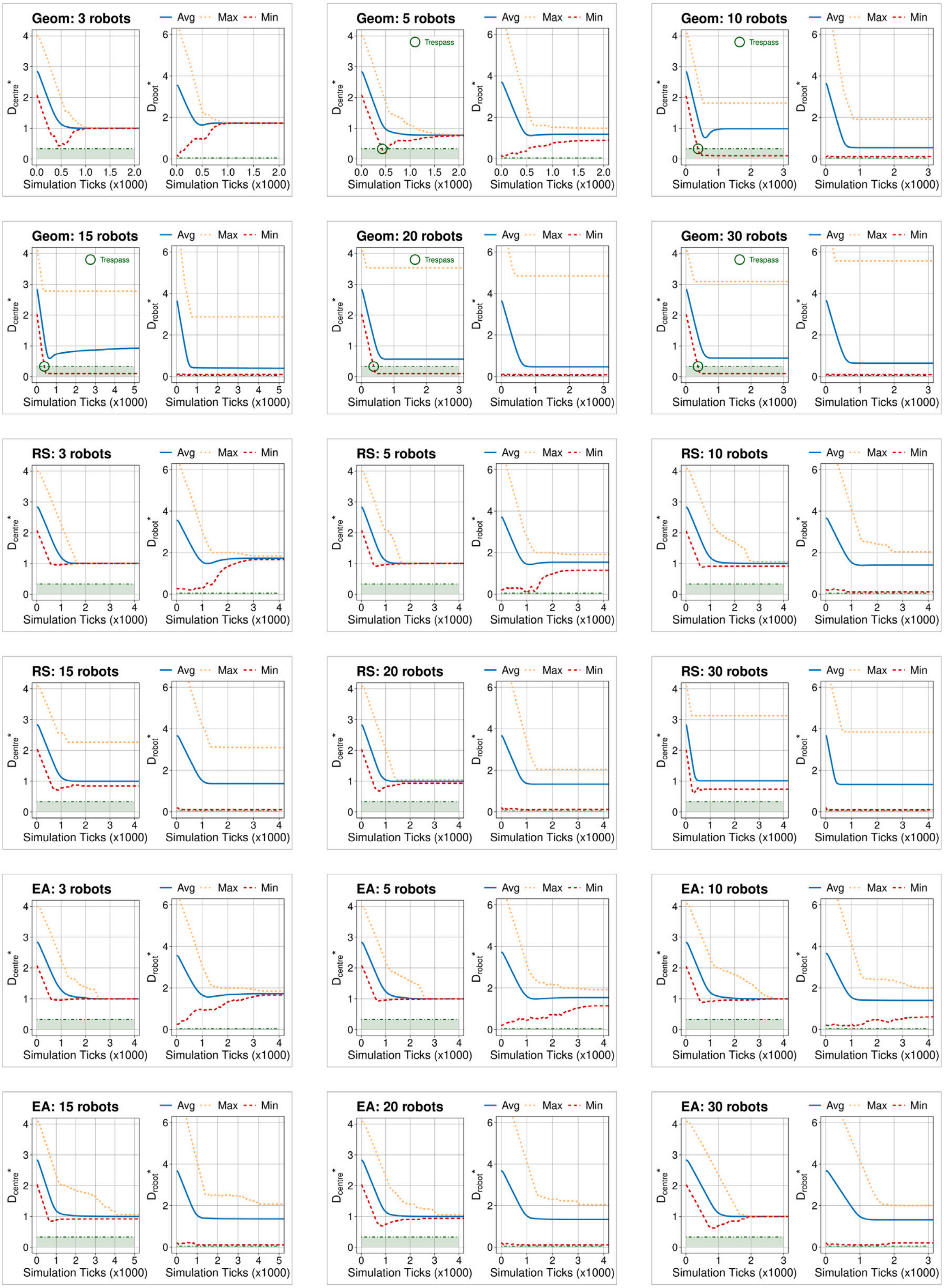


Fig. 9. Average distance of the robots to the central point of interest ($D_{centre}\star$) as well as the average distance between members of the swarm ($D_{robot}\star$). The graphs were calculated from the formations configured using the three approaches tested on 70 unseen scenarios per case study.

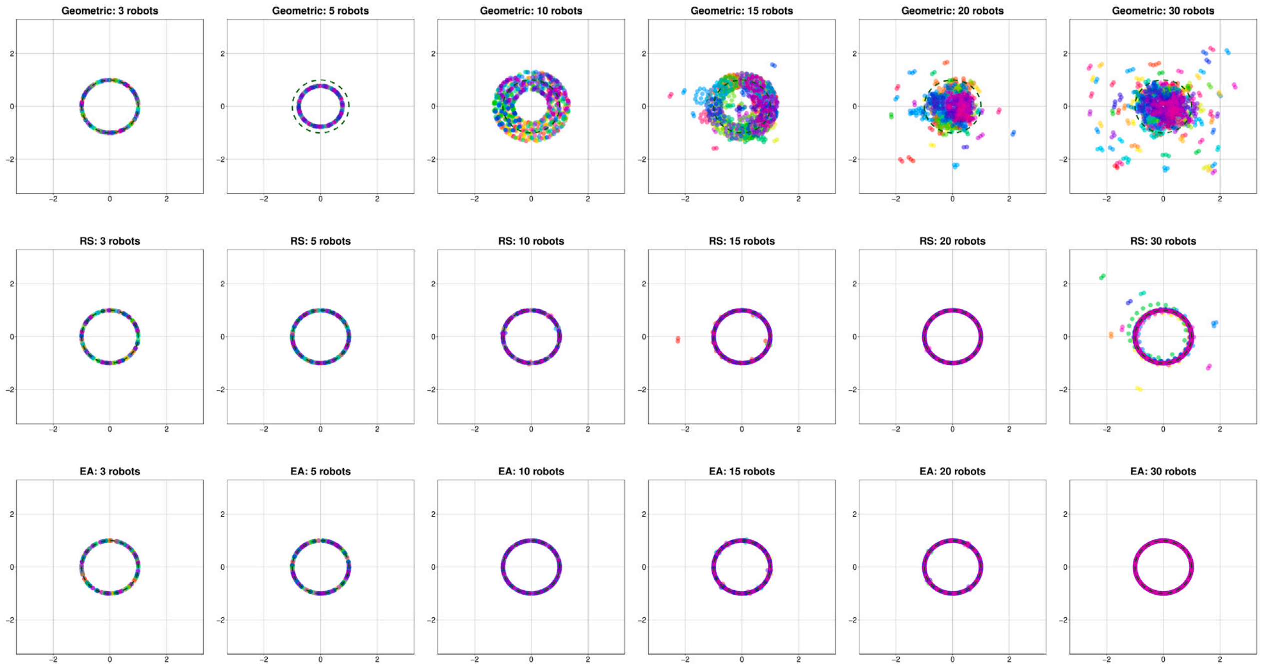


Fig. 10. Final formation positions of robots for 70 scenarios per case study and approach. A dashed circle indicates the desired distance to the centre.

Table 6
Failure and communication loss time table in seconds.

Robots	One failure	Two failures		Three failures			Loss duration
3	103.0	68.7	103.0	34.4	68.7	103.0	10.0
5	206.6	103.3	206.6	67.7	103.3	206.6	10.0
10	203.1	101.5	203.1	67.7	101.5	203.1	10.0
15	192.3	96.2	192.3	64.1	96.2	192.3	10.0
20	202.8	101.4	202.8	100.9	101.4	202.8	10.0
30	162.9	81.5	162.9	54.3	81.5	162.9	10.0

parameter values calculated by the optimisation of 30 scenarios using the EA are not showing overfitting issues and have worked properly in all the tests.

6.3. Fault tolerance

These set of experiments conducted using the ARGoS simulator consisted in evaluating how our formation system reacts when some robots stop broadcasting their beacon due to communication loss or catastrophic failure. We have programmed two types of robot failures in the testing scenarios of the six swarm sizes studied, so that some robots leave the scenario one by one (robot failure) or they suffer a momentary communication loss. The rest of the swarm has to rebuild the formation using the DFA and the current parameterisation, e.g. 27 robots would use the configuration obtained by the EA for 30 robots in order to achieve a successful formation. On the contrary, when there is a communication loss, the affected robot stops moving and the rest of the swarm continues building the formation. Once the communications are re-established, the full swarm would collaborate again and the final positions would be adjusted. We have included the three-failure study for swarms of five and three robots, although in case of unrecoverable failures, the former overlaps with the three-robot swarm study and the latter is not producing any values. Table 6 shows the time table for the programmed failures which were randomly calculated. They are occurring at the same times in both studies for comparison purposes.

Table 7 shows the results obtained from the fault tolerance study. We can see that in the majority of cases the formation was rebuilt as the robots have rearranged themselves in order to keep the desired distance to the centre. The DFA has failed to place the robots in a perfect formation only in 3 scenarios out of the 1190 having at least one permanent robot failure. A similar behaviour can be observed when there are temporal communication losses. As this is a less punitive failure, the formations have been successfully achieved once the swarm has recovered all its members in 1259 scenarios (just one formation failure was observed).

The formation was expected to fail in case studies featuring a higher number of robots since the swarm is still using the same parameters as when it was complete. However, we can see that the DFA had troubles only with a few scenarios (20 and 30 robots) where the swarm iterations are more likely to converge to a non-circular shape (maximum $D_{centre}\star$ much greater than 1.0 or minimum $D_{centre}\star$ much lower than 1.0). A possible solution to achieve 100% successful formations when there are robot failures could consist in calculating a whole set of parameters in advance to cover these situations, or interpolate the values for the case studies already known. All in all, an overall 99.8% success rate is a promising result, especially if we take into account that the DFA have still worked (converged to a stable shape) in all the cases simulating unexpected (and unlikely) robot failures.

Analysing the elapsed times to achieve successful formations, we can see that an extra time was needed in all cases, except for swarms of one and two robots (three-robot case study, one and two permanent failures). It was to be expected as the nominal behaviour of the swarm has been altered by external unexpected factors not occurring in the 0-failure cases. The extra time required increases with the number of failures in almost all the cases, being about 11% on average for both robot failure conditions studied.

Table 7
Results of the fault tolerance study (permanent robot failure and temporal communication loss) for 70 scenarios per case study ($D_{centre} = 1.0$). Times are in minutes and distances are in metres.

Robots	# Failures	Permanent Robot Failure					Temporal Communication Loss				
		D_{centre}★			Formation		D_{centre}★			Formation	
		Min.	Median	Max.	Time	Success	Min.	Median	Max.	Time	Success
3	0	0.998	1.000	1.004	4.00	**100.0%**	0.998	1.000	1.004	4.00	**100.0%**
	1	0.998	1.000	1.001	2.51	**100.0%**	0.998	1.000	1.002	4.21	**100.0%**
	2	0.998	1.001	1.002	2.03	**100.0%**	0.998	1.000	1.002	4.28	**100.0%**
	3	—	—	—	—	—	0.998	1.000	1.002	4.46	**100.0%**
5	0	0.998	1.000	1.002	3.60	**100.0%**	0.998	1.000	1.002	3.60	**100.0%**
	1	0.997	0.998	1.000	4.53	**100.0%**	0.998	1.000	1.002	3.88	**100.0%**
	2	0.994	0.996	0.999	5.21	**100.0%**	0.998	1.000	1.002	3.99	**100.0%**
	3	0.992	1.000	1.001	4.06	**100.0%**	0.998	1.000	1.002	4.08	**100.0%**
10	0	0.997	1.000	1.002	3.49	**100.0%**	0.997	1.000	1.002	3.49	**100.0%**
	1	0.997	0.999	1.001	4.19	**100.0%**	0.998	1.000	1.001	3.92	**100.0%**
	2	0.997	0.999	1.001	4.21	**100.0%**	0.998	1.000	1.001	3.99	**100.0%**
	3	0.997	0.999	1.001	4.28	**100.0%**	0.998	1.000	1.002	3.97	**100.0%**
15	0	0.934	1.001	1.062	3.56	**100.0%**	0.934	1.001	1.062	3.56	**100.0%**
	1	0.922	1.000	1.058	3.95	**100.0%**	0.926	1.000	1.066	3.90	**100.0%**
	2	0.934	1.000	1.067	3.87	**100.0%**	0.917	1.001	1.063	3.83	**100.0%**
	3	0.944	1.000	1.048	3.85	**100.0%**	0.926	1.000	1.064	3.86	**100.0%**
20	0	0.927	0.999	1.056	3.43	**100.0%**	0.927	0.999	1.056	3.43	**100.0%**
	1	0.929	0.999	1.042	3.95	**100.0%**	0.930	0.999	1.044	3.96	**100.0%**
	2	0.918	0.999	1.376	3.91	97.1%	0.939	0.999	1.051	3.88	**100.0%**
	3	0.929	0.999	1.051	3.90	**100.0%**	0.925	0.999	1.048	3.89	**100.0%**
30	0	0.997	0.999	1.002	2.71	**100.0%**	0.997	0.999	1.002	2.71	**100.0%**
	1	0.996	0.999	1.001	3.24	**100.0%**	0.997	0.999	1.002	3.14	**100.0%**
	2	0.875	0.999	1.090	3.24	98.6%	0.996	0.999	1.002	3.15	**100.0%**
	3	0.996	0.999	1.002	3.22	**100.0%**	0.945	0.999	1.500	3.15	98.6%

Table 8
Results of the detection range study for 70 scenarios per case study ($D_{centre} = 1.0$). Times are in minutes and distances are in metres.

Robots	Range: $2 \times D_{centre}$					Range: $3 \times D_{centre}$					Range: $5 \times D_{centre}$				
	D_{centre}★			Formation		D_{centre}★			Formation		D_{centre}★			Formation	
	Min.	Median	Max.	Time	Success	Min.	Median	Max.	Time	Success	Min.	Median	Max.	Time	Success
3	0.998	1.349	3.878	3.73	31.4%	0.998	1.000	3.668	4.20	97.1%	0.998	1.000	1.004	4.00	**100.0%**
5	0.998	1.000	2.618	3.81	87.1%	0.997	1.000	1.002	3.67	**100.0%**	0.998	1.000	1.002	3.60	**100.0%**
10	0.920	1.000	2.422	3.51	94.3%	0.998	1.000	1.002	3.53	**100.0%**	0.997	1.000	1.002	3.49	**100.0%**
15	0.926	1.001	2.387	3.51	97.1%	0.918	1.000	1.058	3.42	**100.0%**	0.934	1.001	1.062	3.56	**100.0%**
20	0.905	0.999	1.313	3.44	97.1%	0.928	0.999	1.044	3.34	**100.0%**	0.927	0.999	1.056	3.43	**100.0%**
30	0.830	0.999	2.111	2.75	95.7%	0.899	0.999	1.919	2.72	95.7%	0.997	0.999	1.002	2.71	**100.0%**
All	0.830	1.000	3.878	5.08	83.8%	0.899	1.000	3.668	5.15	98.8%	0.927	1.000	1.062	5.13	**100.0%**

6.4. Detection range study

In our experiments we have used the beacons from the swarm members and the central object to build the formation. As there are other possibilities for detecting the central object, e.g. images from cameras, we address the study of the DFA behaviour under limited visibility or detection range. The base case used during the optimisation and our tests so far, has a maximum range of 5 metres, i.e. 5 times the formation radius. We have limited the detection range in our scenarios to 2 and 3 metres to obtain the results presented in Table 8.

It can be seen that there is little difference when the maximum detection range is 3 metres. Under this condition the 98.8% of scenarios ended up in a successful formation and only five scenarios failed. On the other hand, when the maximum range is 2 metres, the swarm has struggled to centre the formation around the central object in 68 scenarios (16.2%). This occurs because most of the time, the DFA is controlling the robots without one of the important components to determine the next moving direction, i.e. the attracting/repelling force to/from the centre. It is worth to notice that the DFA still worked on 83.8% of the scenarios under this unexpected, unfavourable situation.

6.5. Scalability study

This section presents a study about the scalability of our proposal, to analyse its limitations in terms of the swarm size. From the computing point of view the code can be run on each robot's CPU. Taking into account the communications, since the system relies on the robot's beacons, the data packets are expected to be very short with no particular data being transmitted apart from the robot's identification. Consequently, the proposed system is unlikely to be sensitive to momentary radio interferences and the formation can be easily repaired if there is a short period of packet loss, as previously studied. Finally, there is a spatial limitation involving the maximum number of robots, which depends on the desired distance to the central object (D_{centre}) in metres. It ultimately defines the length of the circumference (perimeter), i.e. the available space for the robots in formation. Hence, the maximum number of robots is given by Eq. (8), where r_{ep} is the E-Puck2 robot radius and γ is a factor that defines a safe gap around each robot. It can be seen that our experiments are enforcing this limit, i.e. $MAX_{robots} = 44$.

$$MAX_{robots} = \left\lfloor \frac{2\pi \times D_{centre}}{\gamma \times r_{ep}} \right\rfloor, \quad r_{ep} = 0.035, \quad \gamma = 4 \tag{8}$$

6.6. Real world validation

Finally, once we have satisfactorily evaluated our proposal *in silico* using the ARGoS simulator, we wanted to go a step further and also evaluate it *in vivo*. We have set up our experimental environment in the SwarmLab of the FSTM/DCS (University of Luxembourg) as described by the schema in Fig. 11. On the experimentation board we have placed the E-Puck2 (GCtronic, 2023) robots equipped with ArUco markers (Garrido-Jurado et al., 2014) in order to detect the absolute orientation angle of each robot (onboard compass has not provided a

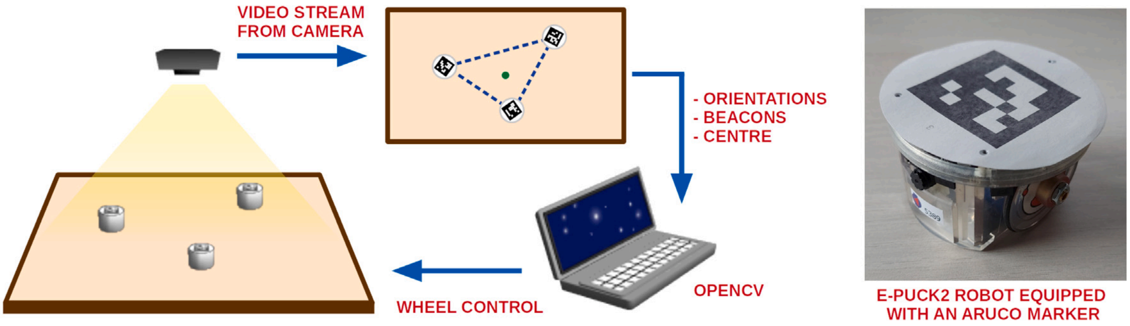


Fig. 11. Experimental environment using E-Puck2 robots, video camera, ArUco markers, and OpenCV.

Table 9
Results of the real world validation using E-Puck2 robots ($D_{centre} = 25.0$). Distances are in centimetres.

Robots	Failures	$D_{robot}\star$			$D_{centre}\star$			Formation
		Min.	Median	Max.	Min.	Median	Max.	
3	0	39.0	42.2	48.1	24.7	24.9	25.1	100.0%
5	0	28.8	38.2	47.8	24.3	24.8	25.3	100.0%
	1	33.6	35.5	49.7	24.3	24.8	25.2	100.0%
	2	40.8	42.7	44.7	24.5	24.7	24.8	100.0%
10	0	14.0	34.9	50.2	24.5	24.9	25.4	100.0%
	1	16.6	37.4	49.8	24.6	24.8	25.3	100.0%
	2	18.4	35.2	50.0	24.3	24.8	25.1	100.0%
	3	21.0	38.8	48.8	24.2	24.8	25.2	100.0%

good precision), and to simulate the communication layer (range and bearing). A zenithal camera streamed the video signal to a computer running our *Environment* software which uses OpenCV (Pulli et al., 2012) to detect each robot position and orientation. The same DFA used in our simulations will control the wheels of the autonomous robots according to the received beacons, while the onboard proximity sensors are used to detect potential collisions. Note that each instance of the DFA corresponding to each robot is running isolated on the computer using OpenCV to simulate the communication layer. The robots only communicate through their beacons in order to build the formation.

Having scaled the parameters optimised by the EA to the new area dimensions, since now the D_{centre} is 25 cm instead of 1 m, we have tested five scenarios per case study consisting of three, five and ten robots. Additionally, we have removed up to three robots per scenario and observed if the formation was to be rebuilt by the remaining robots in the swarm. Table 9 shows the results of the real world validation using E-Puck2 robots. The formations were successfully achieved in all the 40 scenarios, even when some robots were removed. Note that the precision of the measurements is less accurate than when using the simulator although the median of the measured $D_{centre}\star$ values was close to the desired 25.0.

We have plotted the same graphs as in the simulation case, showing in Fig. 12 the convergence of the distances between robots and with respect to the centre. It is especially interesting to observe how close the robots came when building the formation in the ten-robot case study, although no collisions were detected. It seems that a D_{centre} equal to 25 units arranges ten robots so close each other, that they would have been at risk of collision if no avoidance algorithm had been implemented.

Finally, Fig. 13 shows the initial and final positions for five overlapped scenarios per case study where robots belonging to each scenario have been encircled in a different colour for improving clarity. Despite the fact we have tried to cover many possible initial positions, including robots arriving in groups, the DFA has always formed a valid shape (triangle, pentagon, or decahedron). A full video showing each case study and the robot failures is available at https://adars.uni.lu.

7. Conclusion and future work

This article propose a novel distributed control system for a swarm of autonomous robots when executing formation tasks. We have identified the problem parameters, analysed them, and proposed an evolutionary algorithm to optimise their values for six different case studies comprising 30 scenarios each. A systematic method was used to tune the evolutionary algorithm and two competitor techniques were also proposed to evaluate the viability of our proposal. We have tested the six achieved optimal solutions (one per case study) on 70 unseen scenarios each, to obtain successful formations in 100% of cases. A fault tolerance study was also conducted to address the swarm behaviour when some of its members temporarily or permanently disappear. We have found that our Distributed Formation Algorithm was able to rebuild the formation in 99.8% of the fault scenarios tested. A study regarding different detection ranges was also performed where we have observed 83.3% and 98.8% of successful formations for ranges limited to 2 and 3 metres, respectively. Finally, we have evaluated our proposal on 40 scenarios using actual E-Puck2 robots in order to validate the simulation results by scaling the optimal configuration values to the dimensions of our real world scenario. In this case, all the experiments ended in successful formations even in those including robot failures.

We have enforced the system robustness by calculating the optimal parameters for the EA using the well-known irace method, by optimising 30 different scenarios to avoid overfitting, and finally, by testing the best configuration calculated on 420 unseen scenarios where the robots have converged to stable formations in all of them.

We would like to include scenarios with obstacles in future works and test robot formations following a moving central object. This would require adding these types of scenarios to the training set and probably making some modifications to the DFA. New constraints would appear as the robots in the swarm ought to be faster than the object being surrounded. We are also interested in evaluating other metrics such as energy consumption as well as testing other ground robots and heterogeneous swarms. The need of using more than one set of parameters when the robots are different would have to be studied. We are currently working to extend our proposal by adapting our formation algorithm to three-dimensional space using spherical coordinates. We plan to conduct experiments using not only simulations but also quad rotor mini drones. It would require extra computational resources as the scenarios would be bigger and calculations would have to take into account robots' altitude while avoiding collisions and turbulences.

CRediT authorship contribution statement

Daniel H. Stolfi: Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualisation, Supervision. **Grégoire Danoy:** Methodology, Resources, Writing – original draft, Writing – review & editing, Project administration, Funding acquisition.

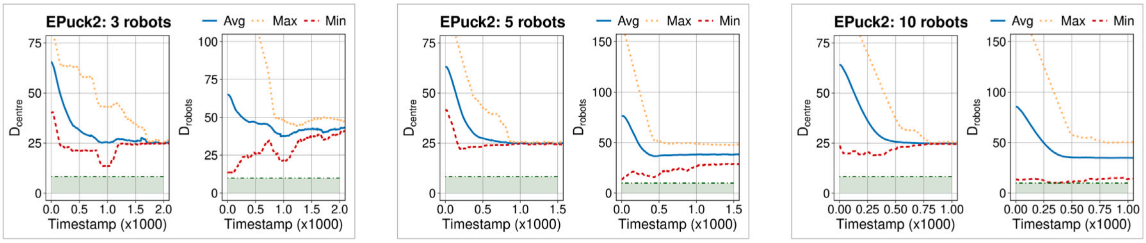
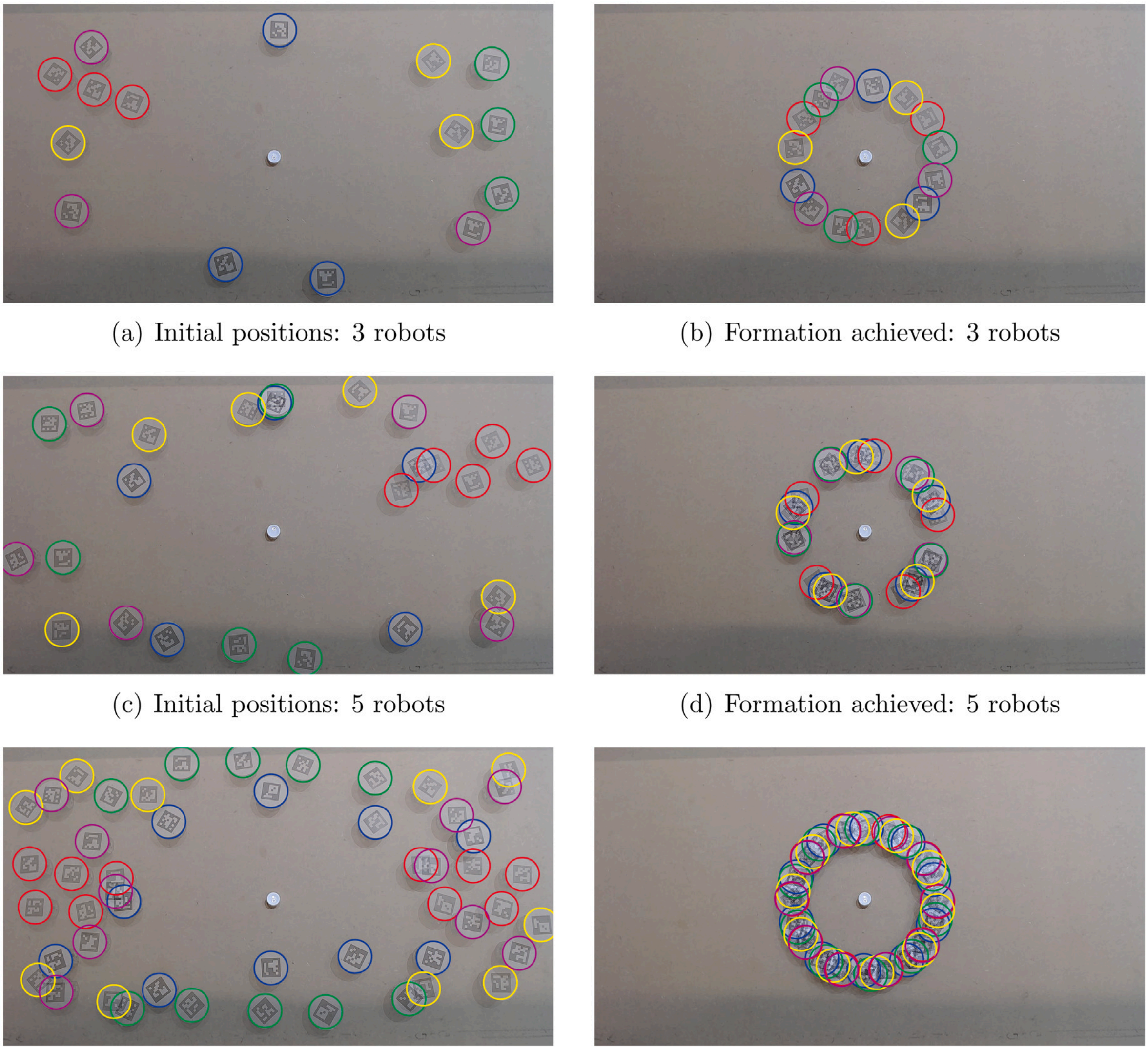


Fig. 12. Average distance of the E-Puck2 robots to the central point of interest ($D_{centre}\star$) as well as the average distance between members of the swarm ($D_{robot}\star$). Five scenarios per case study. All the distances are in centimetres.



(a) Initial positions: 3 robots

(b) Formation achieved: 3 robots

(c) Initial positions: 5 robots

(d) Formation achieved: 5 robots

(e) Initial positions: 10 robots

(f) Formation achieved: 10 robots

Fig. 13. Initial positions and formations achieved using the E-Puck2 robots. Each one of the five overlapped scenarios per case study is highlighted in a different colour.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This work is supported by the Luxembourg National Research Fund (FNR) – ADARS Project, ref. C20/IS/14762457. The experiments presented in this paper were carried out using the SwarmLab facility of the FSTM/DCS and the HPC facilities of the University of Luxembourg (Varrette et al., 2014) – see https://hpc.uni.lu.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.engappai.2023.107501.

References

Ahmed, A.A., Abdalla, T.Y., Abed, A.A., 2015. Path planning of mobile robot by using modified optimized potential field method. Int. J. Comput. Appl. 113, 6–10. http://dx.doi.org/10.5120/19812-1614.

Aldana-Franco, F., Montes-González, F., Ochoa Zezzatti, A., 2021. Acetyl-modulated architecture for evolutionary robotics. Int. J. Comb. Optim. Probl. Inform. 13, 33–49.

Bäck, T., Schwefel, H.-P., 1993. An overview of evolutionary algorithms for parameter optimization. Evol. Comput. 1, 1–23. http://dx.doi.org/10.1162/evco.1993.1.1.1.

Beard, R., Lawton, J., Hadaegh, F., 2001. A coordination architecture for spacecraft formation control. IEEE Trans. Control Syst. Technol. 9, 777–790. http://dx.doi.org/10.1109/87.960341.

Benítez-Hidalgo, A., Nebro, A.J., García-Nieto, J., Oregi, I., Ser, J.D., 2019. jMetalPy: A python framework for multi-objective optimization with metaheuristics. Swarm Evol. Comput. 100598. http://dx.doi.org/10.1016/j.swevo.2019.100598.

Bezcioglu, M.B., Lennox, B., Arvin, F., 2021. Self-organised swarm flocking with deep reinforcement learning. In: 2021 7th International Conference on Automation, Robotics and Applications. ICARA, pp. 226–230. http://dx.doi.org/10.1109/ICARA51699.2021.9376509.

Brust, M.R., Danoy, G., Stolfi, D.H., Bouvry, P., 2021. Swarm-based counter UAV defense system. Discov. Internet Things 1, http://dx.doi.org/10.1007/s43926-021-00002-x.

Campolongo, F., Cariboni, J., Saltelli, A., 2007. An effective screening design for sensitivity analysis of large models. Environ. Model. Softw. 22, 1509–1518. http://dx.doi.org/10.1016/j.envsoft.2006.10.004.

Cardona, G.A., Calderon, J.M., 2019. Robot swarm navigation and victim detection using rendezvous consensus in search and rescue operations. Appl. Sci. 9, http://dx.doi.org/10.3390/app9081702.

Carreon-Ortiz, H., Valdez, F., Castillo, O., 2022. A new discrete mycorrhiza optimization nature-inspired algorithm. Axioms 11, http://dx.doi.org/10.3390/axioms11080391.

Chella, A., Gaglio, S., Mannone, M., Pilato, G., Seidita, V., Vella, F., Zammuto, S., 2023. Quantum planning for swarm robotics. Robot. Auton. Syst. 161, 104362. http://dx.doi.org/10.1016/j.robot.2023.104362.

Chung, S.-J., Ahsun, U., Slotine, J.-J.E., 2009. Application of synchronization to formation flying spacecraft: Lagrangian approach. J. Guid. Control Dyn. 32, 512–526. http://dx.doi.org/10.2514/1.37261.

Cohen, S., Agmon, N., 2021. Recent advances in formations of multiple robots. Curr. Robot. Rep. 2, 159–175. http://dx.doi.org/10.1007/s43154-021-00049-2.

De Jong, K.A., 1975. An Analysis of the Behavior of a Class of Genetic Adaptive Systems (Ph.D. thesis).

Deb, K., 2001. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Inc., USA.

Derrac, J., García, S., Molina, D., Herrera, F., 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. 1, 3–18. http://dx.doi.org/10.1016/j.swevo.2011.02.002.

Dias, P.G.F., Silva, M.C., Rocha Filho, G.P., Vargas, P.A., Cota, L.P., Pessin, G., 2021. Swarm robotics: A perspective on the latest reviewed concepts and applications. Sensors 21, http://dx.doi.org/10.3390/s21062062.

Fazenda, P.V., Lima, P.U., 2007. Non-holonomic robot formations with obstacle compliant geometry. IFAC Proc. Vol. 40, 439–444. http://dx.doi.org/10.3182/20070903-3-FR-2921.00075.

Gao, L., Liu, R., Wang, F., Wu, W., Bai, B., Yang, S., Yao, L., 2019. An advanced quantum optimization algorithm for robot path planning. J. Circuits Syst. Comput. 29, 2050122. http://dx.doi.org/10.1142/s0218126620501224.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., Marín-Jiménez, M., 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognit. 47, 2280–2292. http://dx.doi.org/10.1016/j.patcog.2014.01.005.

GCtronic, 2023. GCtronic – Electronics and mechatronics. URL: https://www.gctronic.com/. (Online; Accessed 17 May 2023).

Goldberg, D.E., Deb, K., 1991. A comparative analysis of selection schemes used in genetic algorithms. Found. Genet. Algorithms 1, 69–93. http://dx.doi.org/10.1016/B978-0-08-050684-5.50008-2.

Hauri, S., Alonso-Mora, J., Breitenmoser, A., Siegwart, R., Beardsley, P., 2014. Multi-Robot Formation Control Via a Real-Time Drawing Interface. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 175–189. http://dx.doi.org/10.1007/978-3-642-40686-7_12.

Issa, B.A., Rashid, A.T., 2019. A survey of multi-mobile robot formation control. Int. J. Comput. Appl. 181, 12–16. http://dx.doi.org/10.5120/ijca2019918651.

Jiang, C., Chen, Z., Guo, Y., 2019. Learning decentralized control policies for multi-robot formation. In: 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. AIM, pp. 758–765. http://dx.doi.org/10.1109/AIM.2019.8868898.

Kennedy, J., 2010. Particle swarm optimization. In: Sammut, C., Webb, G.I. (Eds.), Encyclopedia of Machine Learning. Springer US, Boston, MA, pp. 760–766. http://dx.doi.org/10.1007/{978}-{{0}}-387-30164-8_630.

Khatib, O., 1990. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. Springer New York, New York, NY, pp. 396–404.

Kiełczewski, M., Kowalczyk, W., Krysiak, B., 2022. Differentially-driven robots moving in formation–leader-sfollower approach. Appl. Sci. 12, http://dx.doi.org/10.3390/app12147273.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680. http://dx.doi.org/10.1126/science.220.4598.671, arXiv:arXiv:1011.1669v3.

Li, G., St-Onge, D., Pinciroli, C., Gasparri, A., Garone, E., Beltrame, G., 2018. Decentralized progressive shape formation with robot swarms. Auton. Robots 43, 1505–1521. http://dx.doi.org/10.1007/s10514-018-9807-5.

Li, G., Svogor, I., Beltrame, G., 2019. Long-term pattern formation and maintenance for battery-powered robots. Swarm Intell. 13, 21–57. http://dx.doi.org/10.1007/s11721-019-00162-1.

Liang, X., Wang, H., Liu, Y.-H., Chen, W., Liu, T., 2018. Formation control of nonholonomic mobile robots without position and velocity measurements. IEEE Trans. Robot. 34, 434–446. http://dx.doi.org/10.1109/TRO.2017.2776304.

Lin, S., 1965. Computer solutions of the traveling salesman problem. Bell Syst. Tech. J. 44, 2245–2269.

Lin, J., Miao, Z., Zhong, H., Peng, W., Wang, Y., Fierro, R., 2021. Adaptive image-based leader– follower formation control of mobile robots with visibility constraints. IEEE Trans. Ind. Electron. 68, 6010–6019. http://dx.doi.org/10.1109/tie.2020.2994861.

Liu, Y., Bucknall, R., 2018. A survey of formation control and motion planning of multiple unmanned vehicles. Robotica 36, 1019–1047. http://dx.doi.org/10.1017/S0263574718000218.

Lopez-Gonzalez, A., Ferreira, E.D., Hernandez-Martinez, E.G., Flores-Godoy, J.J., Fernandez-Anaya, G., Paniagua-Contro, P., 2016. Multi-robot formation control using distance and orientation. Adv. Robot. 30, 901–913. http://dx.doi.org/10.1080/01691864.2016.1159143.

López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., Stützle, T., 2016. The irace package: Iterated racing for automatic algorithm configuration. Oper. Res. Perspect. 3, 43–58. http://dx.doi.org/10.1016/j.orp.2016.09.002.

Louste, C., Liegeois, A., 2000. Near optimal robust path planning for mobile robots: The viscous fluid method with friction. J. Intell. Robot. Syst. 27, 99–112. http://dx.doi.org/10.1023/A:1008102230551.

Ma, J., Lu, H., Xiao, J., Zeng, Z., Zheng, Z., 2020. Multi-robot target encirclement control with collision avoidance via deep reinforcement learning. J. Intell. Robot. Syst. 99, 371–386. http://dx.doi.org/10.1007/s10846-019-01106-x.

Makita, S., Wan, W., 2017. A survey of robotic caging and its applications. Adv. Robot. 31, 1071–1085. http://dx.doi.org/10.1080/01691864.2017.1371075.

Mannone, M., Seidita, V., Chella, A., 2023. Modeling and designing a robotic swarm: A quantum computing approach. Swarm Evol. Comput. 79, 101297. http://dx.doi.org/10.1016/j.swevo.2023.101297.

Mastellone, S., Stipanović, D.M., Graunke, C.R., Intlekofer, K.A., Spong, M.W., 2008. Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. Int. J. Robot. Res. 27, 107–126. http://dx.doi.org/10.1177/0278364907084441.

Mezghiche, M.K., Djedi, N., 2020. Quantum genetic algorithm to evolve controllers for self-reconfigurable modular robots. World J. Eng. 17, 427–435. http://dx.doi.org/10.1108/wje-02-2019-0032.

Morris, M.D., 1991. Factorial sampling plans for preliminary computational experiments. Technometrics 33, 161–174. http://dx.doi.org/10.1080/00401706.1991.10484804, publisher: Taylor & Francis.

Oh, K.-K., Park, M.-C., Ahn, H.-S., 2015. A survey of multi-agent formation control. Automatica 53, 424–440. http://dx.doi.org/10.1016/j.automatica.2014.10.022.

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.M., Dorigo, M., 2012. ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intell. 6, 271–295. http://dx.doi.org/10.1007/s11721-012-0072-5.

Pulli, K., Baksheev, A., Kornyakov, K., Eruhimov, V., 2012. Real-time computer vision with opencv. Commun. ACM 55, 61–69.

Queralta, J.P., Mccord, C., Gia, T., Tenhunen, H., Westerlund, T., 2019. Communication-free and index-free distributed formation control algorithm for multi-robot systems. Procedia Comput. Sci. 151, 431–438. http://dx.doi.org/10.1016/j.procs.2019.04.059.

Saeedi, S., Trentini, M., Seto, M., Li, H., 2015. Multiple-robot simultaneous localization and mapping: A review. J. Field Robotics 33, 3–46. http://dx.doi.org/10.1002/rob.21620.

Stolfi, D.H., Danoy, G., 2022. Optimising autonomous robot swarm parameters for stable formation design. In: Proceedings of the Genetic and Evolutionary Computation Conference. GECCO '22, Association for Computing Machinery, New York, NY, USA, pp. 1281–1289. http://dx.doi.org/10.1145/3512290.3528709.

Stolfi, D.H., Danoy, G., 2023. Design and analysis of an E-Puck2 robot plug-in for the ARGoS simulator. Robot. Auton. Syst. 164, 104412. http://dx.doi.org/10.1016/j.robot.2023.104412.

Sui, Z., Pu, Z., Yi, J., Wu, S., 2021. Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration. IEEE Trans. Neural Netw. Learn. Syst. 32, 2358–2372. http://dx.doi.org/10.1109/TNNLS.2020.3004893.

Talbi, E.-G., 2013. A Unified Taxonomy of Hybrid Metaheuristics with Mathematical Programming, Constraint Programming and Machine Learning. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 3–76. http://dx.doi.org/10.1007/978-3-642-30671-6_1.

Varrette, S., Bouvry, P., Cartiaux, H., Georgatos, F., 2014. Management of an academic HPC cluster: The UL experience. In: 2014 International Conference on High Performance Computing & Simulation. HPCS, IEEE, Bologna, Italy, pp. 959–967. http://dx.doi.org/10.1109/HPCSim.2014.6903792.

Xie, J., Zhou, R., Liu, Y., Luo, J., Xie, S., Peng, Y., Pu, H., 2021. Reinforcement-learning-based asynchronous formation control scheme for multiple unmanned surface vehicles. Appl. Sci. 11, http://dx.doi.org/10.3390/app11020546.

Zhu, K., Zhang, T., 2021. Deep reinforcement learning based mobile robot navigation: A review. Tsinghua Sci. Technol. 26, 674–691. http://dx.doi.org/10.26599/TST.2021.9010012.