Swarm and Evolutionary Computation 81 (2023) 101337

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Automated pattern generation for swarm robots using constrained multi-objective genetic programming

Zhun Fan [a,b,c], Zhaojun Wang [a], Wenji Li [a], Xiaomin Zhu [e,f], Bingliang Hu [g], An-Min Zou [a], Weidong Bao [f], Minqiang Gu [d], Zhifeng Hao [d], Yaochu Jin [h,*]

[a] Department of Electronic Engineering, Shantou University, Shantou 515063, China
[b] International Cooperation Base of Evolutionary Intelligence and Robotics, Shantou 515063, China
[c] Key Laboratory of Intelligent Manufacturing Technology (Shantou University), Ministry of Education, Shantou 515063, China
[d] College of Science, Shantou University, Shantou 515063, China
[e] Strategic Assessments and Consultation Institute, Academy of Military Sciences, Beijing 100091, China
[f] College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
[g] Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710000, China
[h] Faculty of Technology, Bielefeld University, Bielefeld 33619, Germany

ARTICLE INFO

Keywords:
Gene regulatory network (GRN)
Entrapping pattern generation
Self-organization
Constrained multi-objective genetic programming (CMOGP)

ABSTRACT

Swarm robotic systems (SRSs), which are widely used in many fields, such as search and rescue, usually comprise a number of robots with relatively simple mechanisms collaborating to accomplish complex tasks. A challenging task for SRSs is to design local interaction rules for self-organization of robots that can generate adaptive patterns to entrap moving targets. Biologically inspired approaches such as gene regulatory network (GRN) models provide a promising solution to this problem. However, the design of GRN models for generating entrapping patterns relies on the expertise of designers. As a result, the design of the GRN models is often a laborious and tedious trial-and-error process. In this study, we propose a modular design automation framework for GRN models that can generate entrapping patterns. The framework employs basic network motifs to construct GRN models automatically without requiring expertise. To this end, a constrained multi-objective genetic programming is utilized to simultaneously optimize the structures and parameters of the GRN models. A multi-criteria decision-making approach is adopted to choose the preferred GRN model for generating the entrapping pattern. Comprehensive simulation results demonstrate that the proposed framework can obtain novel GRN models with simpler structures than those designed by human experts yet better performance in complex and dynamic environments. Proof-of-concept experiments using e-puck robots confirmed the feasibility and effectiveness of the proposed GRN models.

1. Introduction

In general, swarm robotic systems [1,2] (SRSs) are composed of numerous robots with relatively simple mechanisms. The robots collaborate to execute a complex task that is impossible for a single robot to accomplish. A challenging task for SRSs is target entrapping, which typically requires swarm robots to cooperatively entrap multiple targets [3]. Target entrapping can be extended to several potential applications including gas leak detection [4], search and rescue [5,6], deployment of sensor networks [7,8], convoy/escort missions [9], and area/border coverage [10].

Existing entrapping control models can be generally classified into five categories: leader–follower structure, virtual structure, behavior-based approach, reinforcement learning approach, and biologically inspired approach.

In the leader–follower structure, followers track one or more leaders and maintain a specified geometric relationship [11,12]. In most related studies, swarm robots were required to generate predefined patterns. For example, Han et al. [13] programmed followers to maintain a regular polygonal entrapping pattern while entrapping a moving leader (target). Yu et al. [14] applied the leader–follower structure to design a dynamic control law that allowed all robots to entrap a given stationary target while remaining evenly spaced along the circumference of a circle. In real-world applications, followers should

* Corresponding author.
E-mail address: yaochu.jin@uni-bielefeld.de (Y. Jin).

https://doi.org/10.1016/j.swevo.2023.101337
Received 6 May 2022; Received in revised form 15 May 2023; Accepted 17 May 2023
Available online 27 May 2023
2210-6502/

be able to generate time-varying entrapping patterns while tracking one or more leaders. Therefore, Yang et al. [15] proposed a control method for the leader–follower structure to generate an adaptive entrapping pattern. However, this adaptive entrapping pattern can only be scaled from a fixed polygonal pattern, and the flexibility of this method is insufficient.

In a virtual structure, all the robots share a rigid geometric relationship to perform an entrapping task. For example, Kawakami et al. [16] and Sato et al. [17] applied a circular virtual structure to entrap a target. Barnes et al. [9] defined an elliptical entrapping shape, which provided more flexibility for shape transformation than a circle entrapping shape. Rezaee et al. [18] studied the problem of entrapping control of robots based on a virtual structure, and the designed entrapping shape could adapt to changes in the number of robots, but the shape was limited to regular polygons. In these studies [9,16–18], the robots have to maintain fixed and predefined geometric shapes as the targets moved, which prevented them from adapting to the environment.

The behavior-based approach allows swarm robots to complete an entrapping task through a variety of predefined behaviors (e.g., collision avoidance, obstacle avoidance, target searching, and maintaining a desired position in a formation). For example, Antonelli et al. [19] predefined a behavior matrix and sorted the behaviors according to a priority order. The swarm robots entrapped a target based on this matrix. A shortcoming of this method is that setting the priority order when the number of behaviors is large becomes difficult. Phung et al. [20] selected an optimal combination of behaviors using a trial-and-error method to achieve an entrapping task. In behavior-based methods, the behavior of swarm robots must be predefined. The multiple behaviors required of a robot may lead to contradictory commands that can prevent the robots from converging into the desired formation.

The reinforcement learning approach enables SRSs to learn an optimal entrapment strategy using reward strategies. For example, Ma et al. [21] proposed a deep reinforcement learning framework for entrapping pattern generation. Wang et al. [22] proposed a reinforcement learning method for solving multi-pursuer versus single superior evader games. Park et al. [23] established a co-evolution framework for predators and prey to allow multiple agents to learn good policies through deep reinforcement learning. In these studies [21–23], it is often assumed that the escape strategy of the evader target is known in advance, which is unrealistic in real-world environments. Another limitation is that the large number of uncertainties generated from real-world environments make the reward mechanism difficult to design. In addition, in the reinforcement learning approach, the size of the learning space increases exponentially with the number of agents, causing the “curse of dimensionality” problem.

In these methods, SRSs typically use regular entrapping shapes (e.g. circles [14,17], ellipses [24], and regular polygons [13]) to entrap a single target. However, in real-world applications, the number and distribution of targets may vary, and the environment around the targets may change dynamically. This makes arbitrary entrapping patterns more desirable than regular patterns because they are more flexible and adaptive to various dynamic target scenarios [25,26]. A fixed or regular shape with limited changes cannot satisfy the need for the dynamic entrapping of multiple targets. Adaptively generating/transforming entrapping patterns in a dynamic target scenario and controlling robots to form such patterns for flexible entrapments remains a challenge.

Biologically inspired approaches (e.g., morphogen, chemotaxis, and gene regulatory network (GRN) models) [27] provide promising and flexible alternatives for generating adaptive and robust patterns amid unknown environmental changes [28,29]. The morphogen and chemotaxis models consider a robot as a cell that can generate an exponentially decreasing morphogen gradient field [30]. The gradient field can cause its neighborhood robots to move according to the concentration difference, which generates aggregation behavior [31]. However, even though the morphogen and chemotaxis models can guide robots to approach a target, they cannot generate adaptable and changing entrapping patterns for swarm robots to encircle the target. A GRN model treats targets and obstacles as cells to produce two types of proteins and modulates the concentration of these two types of proteins to generate an entrapping pattern.

For example, Jin et al. [32] proposed a hierarchical gene regulatory network (H-GRN) model to entrap dynamic targets. The H-GRN model consists of two layers: the top layer generates an entrapping pattern, and the bottom layer organizes the swarm robots to move into the desired pattern while avoiding obstacles. Peng et al. [33] presented a modified GRN model for enclosing multiple targets in an environment with obstacles. The top layer of the model uses implicit interpolation functions to generate entrapping patterns. The position information of the targets and obstacles was used as focal points of an implicit interpolation function to adjust the patterns. Therefore, there was no overlap between the generated patterns and obstacles. Using a similar approach, Zhang et al. [25] applied an implicit radial basis function to generate entrapping patterns in the top layer of a GRN model. Although these methods could [25,33] generate irregular entrapping patterns, they required large number of feature points determining the shapes of patterns in advance, which cannot be easily obtained from the environment. Oh et al. [34] introduced an evolving H-GRN (named EH-GRN) to generate adaptive patterns that entrap targets in environments with obstacles. Obstacle information is input to the top layer to generate an expected pattern to avoid obstacles. Yuan et al. [35] added an inter-layer based on the H-GRN for obstacle avoidance. They considered the target information in the top layer of the GRN and utilized the obstacle information in the inter-layer. In these studies, GRN models [34,35] have fixed and complex structures predefined by human experts. Therefore, the design automation of a simple and efficient GRN model to generate flexible entrapping patterns under various scenarios remains a challenge.

A GRN model can be automatically assembled from a predefined set of building blocks [36–38]. The design of the GRN model can be first formulated as an automatic modular design problem. Then, we can optimize the structure and parameters of the GRN models [39–41] using evolutionary algorithms [42]. In this paper, we propose a framework that uses constrained multi-objective genetic programming (CMOGP), which combines genetic programming (GP), push and pull search (PPS) [43], and differential evolution (DE) [44]. For CMOGP, GP is responsible for expressing and evolving the GRN models. PPS is applied to search for the optimal set of design candidates satisfying constraints to balance the complexity and performance of the evolved GRN models, and DE is used to optimize the parameters of the GRN models. CMOGP can simultaneously optimize the structure and parameters of GRN models. Novel GRN models with simpler structures than those designed by human experts can be extracted by investigating the structures of evolved models in the optimal Pareto set. More importantly, these evolved models maintained satisfactory performance when transferred to a new application scenario for which the models have never been trained before. It is notable that the proposed design automation framework does not require predefined GRN structures. Instead, a set of basic network motifs are employed as building blocks for the proposed method to reconfigure themselves into functional GRN models in an automated manner, which provides a new method of generating functional GRN models for the applications under investigation.

The main contributions of this study are as follows:

1. A modular design automation framework is proposed to obtain GRN models for swarm robots to generate entrapping patterns, which eliminates the need of human expertise and trial and error. The models are verified through comprehensive experiments.

2. A new algorithm named CMOGP is proposed to optimize both the structures and parameters of the GRN model. Compared with evolving H-GRN (EH-GRN) [34], which can only optimize the parameters of a GRN model, the proposed method can achieve better performance.
3. The automatically obtained GRN models not only outperform those designed by human experts in challenging environments but are also transferable to new application scenarios. The evolved models perform significantly better than human-designed GRN models in new environments.

The remainder of this paper is organized as follows. Section 2 presents the problem formulation and fundamentals of the GRNs. In Section 3, a design automation framework for entrapping pattern generation using the CMOGP is proposed. In Section 4, the simulation and physical experimental results for various challenging application scenarios are discussed to verify the robustness of the framework. In Section 5, the conclusions and future research directions are discussed.

2. Background and problem formulation

2.1. GRN models developed for SRSs

According to some studies [27,28], biologically inspired swarm control methods can be divided into two categories: macroscopic and microscopic. The former refers to methods that are inspired by animal behaviors, such as flying bird flocks or swimming fish schools. The latter represents a less well-known group inspired by the collective movement of microscopic individuals, such as cells, in which GRN plays a pivotal role. GRN is a model of the interactions between genes and gene products that describes the dynamics of gene expression, which plays a central role in biological morphogenesis. Biological morphogenesis can be viewed as a self-organizing process in which populations of cells move autonomously to their destinations, governed by GRN and cell-to-cell interactions. Under the GRN mechanism, each cell releases proteins around it according to certain rules, and the protein concentration decays with spatial distance. Each point in space has a corresponding protein concentration value, and we refer to this concentration space region as the concentration field. Motion control can then be performed based on the gradient properties of the concentration field, as shown in Fig. 1A. In morphogenetic SRSs, each robot is analogous to a single cell. The position of the robot can be transformed into the protein concentration value of the position in the concentration field using the GRN model, as shown in Fig. 1B. In addition, the concentration contours around the target are used as candidate entrapping patterns (satisfying the requirement that the minimum distance between the contour line and the target should surpass the predefined safe distance). According to the gradient properties of the concentration field, each robot moves towards the desired position and eventually remains in the entrapping pattern. A collective encirclement behavior will emerge out of the SRS.

2.2. Problem statement

This study aims to develop a method for entrapping dynamic targets with swarm robots under environmental constraints. The task consists of entrapping pattern generation and formation. In pattern generation, entrapping patterns are generated to encircle the targets while avoiding the obstacles. In pattern formation, the robots deploy themselves towards the generated patterns to accomplish the entrapping task.

2.3. Assumptions

The following basic assumptions are made to implement the proposed design automation framework for pattern generation:

1. The base station contains an adequate number of robots to complete the pattern generation task. In other words, an adequate number of robots can be summoned from the command center to generate entrapping patterns.
2. The SRS uses a motion-capture device with global positioning capability to achieve the entrapping task.
3. The maximum speed of each robot is faster than that of the targets.

2.4. Evaluation metrics

This study employs two evaluation metrics [25] to measure the performance of the proposed method in the entrapping task: (1) the convergence error (C_e), which denotes the entrapment accuracy, and (2) the distributed variance (D_v), which denotes the evenness of the swarm robots deployed in the entrapping pattern.

1. Convergence error (C_e)

$$C_e(\boldsymbol{g}) = \left(\sum_{t=1}^{T}\sum_{i=1}^{n} d_{\min}(f(g_i), i, t)\right)/T \tag{1}$$

   where g_i represents the position of the ith robot. $\boldsymbol{g} = (g_1, g_2, \ldots, g_n)^T$ represents the swarm robot. $f(g_i)$ is an implicit function for generating an entrapping pattern, for example $f(g_i) : g_{i,x}^2 + g_{i,y}^2 - 1 = 0$ is a unit circle. $d_{\min}(f(g_i), i, t)$ is the shortest distance between the ith robot and the expected entrapping pattern in the tth time step. C_e is zero if all the swarm robots are distributed exactly on the pattern. n denotes the number of swarm robots. T denotes the total time step.
2. Distributed variance (D_v)

$$D_v = \sum_{t=1}^{T}\left(1 - \sum_{i=1}^{m_t}\frac{N_{i,t}}{m_t} + \sum_{i=1}^{n}\frac{(d_{i,t}^{\min} - \overline{d_t^{\min}})^2}{n}\right)/T \tag{2}$$

   where m_t indicates that the entrapping pattern is uniformly divided into m sub-patterns in the tth time step. $N_{i,t}$ is unity if there is a robot on the ith part in the tth time step; otherwise, it is zero. $1 - \sum_{i=1}^{m_t}\frac{N_{i,t}}{m_t}$ is zero if all the swarm robots are located on the m sub-patterns in the tth time step. $d_{i,t}^{\min}$ is the shortest distance between the ith robot and its neighbors in the tth time step. $\overline{d_t^{\min}}$ is the mean of all $d_{i,t}^{\min}$ in the tth time step. n denotes the number of swarm robots. T denotes the total time step.

3. Automated GRN design for entrapping pattern generation

In this section, a CMOGP-based design automation framework for entrapping pattern generation is presented. Furthermore, ten predefined basic network motifs as well as fitness functions are discussed.

3.1. Basic network motifs

A fundamental step in the design automation framework is to define a few basic network motifs as the building blocks. Recent systems biology research revealed the occurrence of network module interconnections in real complex networks [45]. Bowers et al. [46] conducted a logic analysis of phylogenetic profiles to discover the triplets of proteins, the presence or absence of which obeys certain logic relationships. These relationships are frequently found in the GRNs of multicellular organisms. Inspired by these findings, ten predefined network motifs, $Positive$, $Negative$, AND, $NAND$, OR, NOR, $ANDN$, ORN, XOR, and $XNOR$, are utilized as the basic network motifs to construct GRNs, as shown in Table 1, where θ and k are regulatory

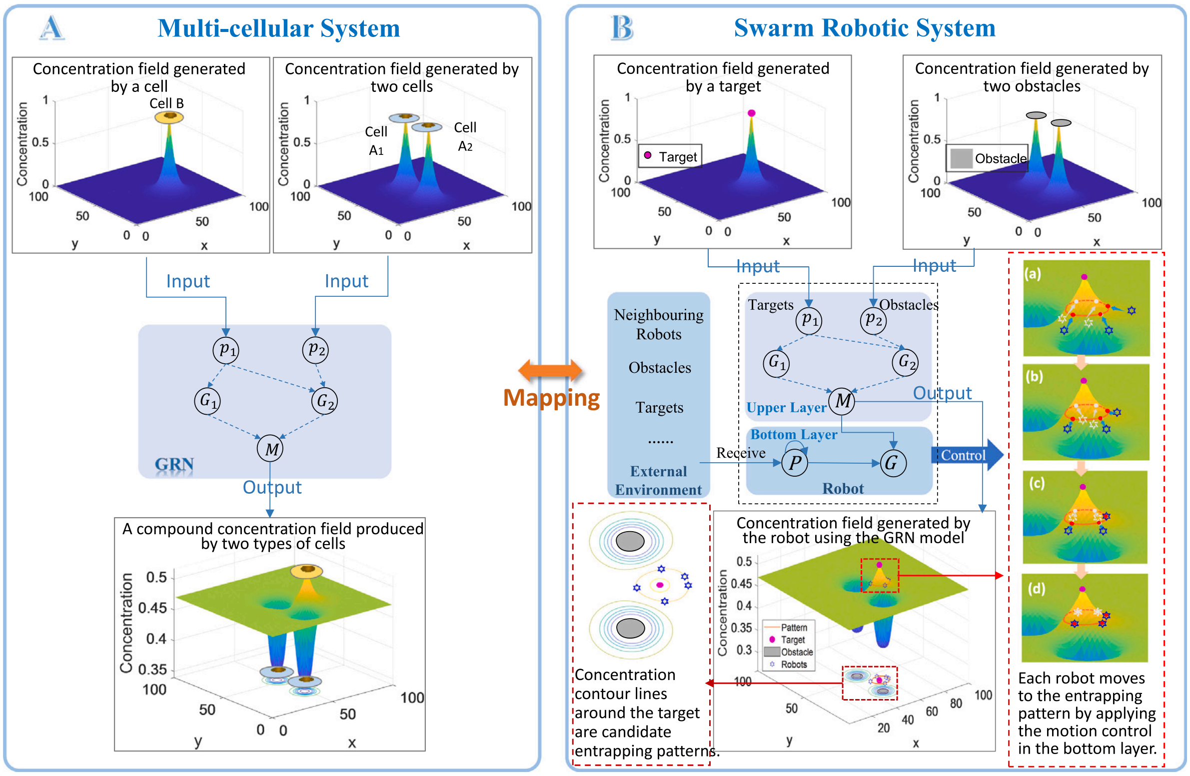


Fig. 1. Mapping relationship between swarm robotic system and multi-cellular system.

parameters and scale factors of gene expression, respectively. Fig. 2 illustrates the tree structure of the GRN through an example. In this figure, the individual is represented by the function set and the terminal set, where the terminal set includes environmental inputs (p_1 and p_2 are the positions of targets and obstacles to establish corresponding concentration fields), and the function set includes all or a subset of the ten predefined network motifs. In this example, the GRN model is

$$\frac{dG_1}{dt} = -G_1 + sig(p_1 + p_2, \theta_1, k) \tag{3}$$

$$\frac{dG_2}{dt} = -G_2 + sig(p_1, \theta_2, k) \tag{4}$$

$$\frac{dG_3}{dt} = -G_3 + sig(G_1 * G_2, \theta_2, k) \tag{5}$$

where G_1 fuses the concentration fields from p_1 and p_2, and G_2 fuses the concentration field from p_1. G_3 fuses the concentration fields from G_1 and G_2, which corresponds to the generation of entrapping patterns. Specifically, the concentration field components of the targets and obstacles are fused by G_1, G_2, and G_3, and then closed contour lines of concentration values are obtained in the G_3 concentration field. The concentration contours around the target are used as candidate entrapping patterns.

3.2. Fitness function

A simpler GRN model is more explainable and preferable. The fewer the nodes in a GRN model, the less complex it is. Therefore, the number of nodes in the GRN model is used to define one of the fitness functions as follows:

$$f_1 = node(\mathcal{M}) \tag{6}$$

where $\mathcal{M}$ is a GRN model and $node(\mathcal{M})$ is the number of its nodes.

An entrapping pattern should enable the robots to entrap the targets without colliding with them. The formulated entrapping pattern should not be too far away from or too close to the targets (to avoid collision with the targets). Furthermore, swarm robots cannot collide with

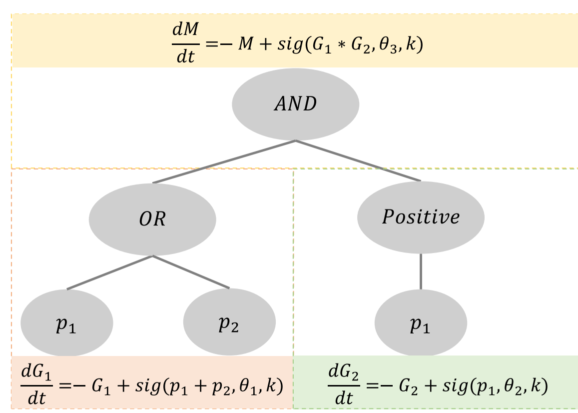


Fig. 2. Example of a GRN tree. AND, OR, and $Positive$ are subsets of the ten predefined network motifs. p_1 and p_2 are the positions of targets and obstacles respectively to establish corresponding concentration fields.

obstacles in the environment. Therefore, the second fitness function is defined as follows:

$$f_2 = \Sigma_{i=1}^{N_p} \Sigma_{j=1}^{N_t} \frac{sig(d_{ij}^{pt}, d_{max}, k_1) + sig(d_{min}, d_{ij}^{pt}, k_2)}{N_p N_t} \tag{7}$$

$$sig(x, \theta, k) = \frac{1}{1 + e^{-k(x-\theta)}} \tag{8}$$

where d_{min} and d_{max} are the allowed minimum and maximum distances between the swarm robots and targets, respectively. In Eq. (7), N_p and N_t represent the number of swarm robots and targets, respectively, and d_{ij}^{pt} is the distance from the ith swarm robot to the jth target. Slope values (k_1 and k_2) for the sigmoid function in Eq. (7) are both set to 1 in this work.

Table 1
Definitions of ten predefined network motifs.

Network motifs	Definition	Functional equation
Positive	Gene x activates gene y. In other words, gene x provides positive feedback to gene y.	$\begin{cases} \frac{dy}{dt} = -y + sig(x, \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$
Negative	Gene x inhibits gene y. That is, gene x provides negative feedback to gene y.	$\begin{cases} \frac{dy}{dt} = -y + 1 - sig(x, \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$
AND	Gene y is present if and only if both gene x_1 and gene x_2 are present.	$\begin{cases} \frac{dy}{dt} = -y + sig(x_1 * x_2, \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$
NAND	Gene y is present if and only if gene x_1 or gene x_2 is absent.	$\begin{cases} \frac{dy}{dt} = -y + sig((1-x_1) + (1-x_2), \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$
OR	Gene y is present if and only if gene x_1 or gene x_2 is present.	$\begin{cases} \frac{dy}{dt} = -y + sig(x_1 + x_2, \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$
NOR	Gene y is present if and only if gene x_1 and gene x_2 are present.	$\begin{cases} \frac{dy}{dt} = -y + sig((1-x_1) * (1-x_2), \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$
ANDN	Gene y is present if and only if gene x_1 is present and gene x_2 is absent.	$\begin{cases} \frac{dy}{dt} = -y + sig(x_1 * (1-x_2), \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$
ORN	Gene y is present if and only if gene x_1 is present or gene x_2 is absent.	$\begin{cases} \frac{dy}{dt} = -y + sig(x_1 + (1-x_2), \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$
XOR	Gene y is present if and only if both genes x_1 and x_2 are present or both are absent.	$\begin{cases} \frac{dy}{dt} = -y + sig(x_1 * (1-x_2), \theta, k) \\ \qquad +sig((1-x_1) * x_2, \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$
XNOR	Gene y is present if and only if one of either gene x_1 or x_2 is present.	$\begin{cases} \frac{dy}{dt} = -y + sig((x_1 * x_2), \theta, k) \\ \qquad +sig((1-x_1) * (1-x_2), \theta, k) \\ sig(x, \theta, k) = \frac{1}{1+e^{-k(x-\theta)}} \end{cases}$

The effects of obstacles and minimal safe distance between neighboring robots on the pattern must be considered during pattern generation. Two types of constraints are defined as follows.

$$d_i^{\text{obs}} \geq d_{\text{min}}^{\text{obs}} \tag{9}$$

$$d_{ij}^{\text{robot}} \geq d_{\text{min}}^{\text{robot}} \tag{10}$$

where Eq. (9) assures that the robots do not collide with the obstacles. Eq. (10) assures that two neighboring robots do not collide. $d_{\text{min}}^{\text{obs}}$ is the minimum allowed distance between the swarm robots and obstacles. d_i^{obs} is the distance from the ith robot to the obstacle. d_{ij}^{robot} is the distance from the ith robot to the jth robot. $d_{\text{min}}^{\text{robot}}$ denotes the minimum safe distance between neighboring robots.

3.3. Design automation framework for entrapping pattern generation based on CMOGP

Because the design of a GRN model is usually a trial-and-error process, it is largely dependent on the experience and intuition of human experts. However, this process does not guarantee the feasibility of the GRN model for a specific application. Fig. 3 illustrates the general structure of the proposed design automation framework for generating entrapping patterns, which is divided into three parts: the construction of the hierarchical GRN model, the automated design of its upper layer using CMOGP, and its application in various scenarios. The proposed design automation framework is summarized as follows:

(1) Construction of a hierarchical GRN model: The proposed hierarchical GRN model is shown in Fig. 3. It is divided into two layers: an entrapping pattern generation layer and an entrapping pattern formation layer.

The upper layer of the hierarchical GRN is responsible for generating entrapping patterns that can adapt to the changes in the environment and targets. Unlike relying on the experience and intuition of human experts, this layer is designed using CMOGP. The locations of the targets (p_1) and obstacles (p_2) are translated into an integrated morphogen gradient space in which the GRN model is automatically reconfigured with predefined network motifs. Specifically, the concentration field M fuses the concentration fields of targets and obstacles, from which closed contour lines of concentration values are obtained and transmitted to the lower layer.

The bottom layer is responsible for forming entrapping patterns that guide swarm robots to approach the pattern generated in the upper layer. This layer utilizes the dynamics of swarm robot motion, represented by two vectors ($\boldsymbol{G}$ and $\boldsymbol{P}$) [34]. $\boldsymbol{P}$ represents the internal state of the robots and is used to receive the position information from neighboring robots, targets, and obstacles to prevent collisions. If the distance between the robot and the target is less than the safe distance, $\boldsymbol{P}$ can also control the robot to move away from the target. On the other hand, $\boldsymbol{G}$ is dynamically adjusted based on the state of $\boldsymbol{P}$ and the entrapping pattern, ensuring that the robots move towards the entrapping pattern. The regulatory mechanism of the bottom layer is as follows:

$$\frac{d\boldsymbol{G}_i}{dt} = -\boldsymbol{G}_i + \boldsymbol{P}_i + \boldsymbol{c}_i \tag{11}$$

$$\frac{d\boldsymbol{P}_i}{dt} = -\boldsymbol{P}_i + \boldsymbol{D}_i + \boldsymbol{T}_i + \boldsymbol{O}_i \tag{12}$$

where $i \in 1, \ldots, N$ is the index of ith robot, and N is the number of robots. $\boldsymbol{G}_i$ and $\boldsymbol{P}_i$ are the 2-D position and internal state vectors of the ith robot, respectively. $\boldsymbol{c}_i$ guides the ith robot towards the entrapping pattern, which is defined as follows:

$$\boldsymbol{c}_i = -\frac{\boldsymbol{R}_i - \boldsymbol{C}_i}{\|\boldsymbol{R}_i - \boldsymbol{C}_i\|} \tag{13}$$

where $\boldsymbol{C}_i$ represents the closest position of the entrapping pattern to the ith robot. $\boldsymbol{R}_i$ denotes the positions of the ith robot. In Eq. (12), $\boldsymbol{D}_i$

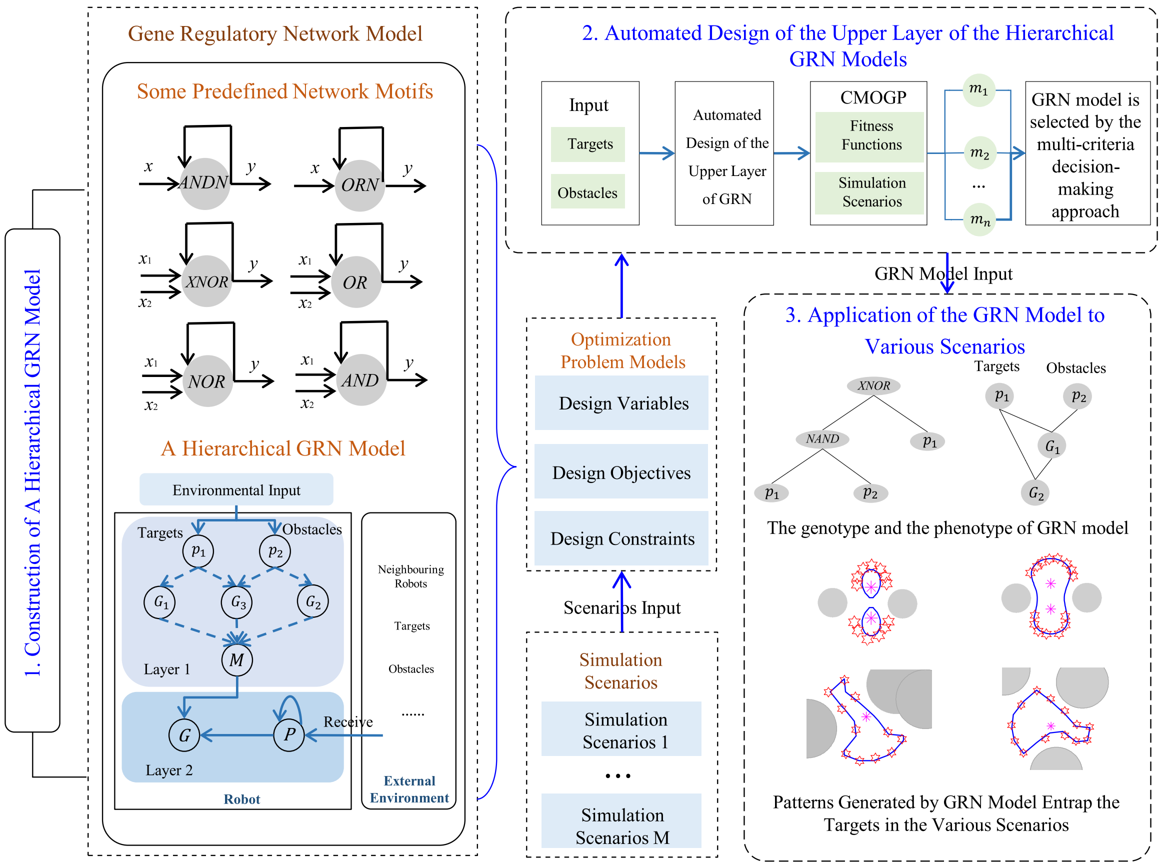


Fig. 3. Diagram of the proposed design automation framework for generating entrapping patterns.

assures that the ith robot does not collide with its neighboring robots.

$$D_i = \sum_{j=1}^{n_i} D_i^j \tag{14}$$

$$D_i^j = \frac{R_i - R_j}{\|R_i - R_j\|} \tag{15}$$

where n_i is the number of neighboring robots of the ith robot. D_i^j is a vector from the ith robot to the jth robot. In addition, the roles of T_i and O_i are to prevent the ith robot from colliding with targets and obstacles, respectively. They are defined as follows:

$$T_i = \sum_{j=1}^{t_i} T_i^j \tag{16}$$

$$T_i^j = \begin{cases} -\frac{R_i - T_j}{\|R_i - T_j\|}, & \|R_i - T_j\| > d_{\max} \\ 0, & d_{\min} \le \|R_i - T_j\| \le d_{\max} \\ \frac{R_i - T_j}{\|R_i - T_j\|}, & \|R_i - T_j\| < d_{\min} \end{cases} \tag{17}$$

$$O_i = \sum_{j=1}^{o_i} O_i^j \tag{18}$$

$$O_i^j = \frac{R_i - O_j}{\|R_i - O_j\|} \tag{19}$$

where t_i and o_i are the numbers of targets and obstacles, respectively. T_i is the sum of direction vectors to guide ith robot to avoid colliding with the targets. O_i is the sum of direction vectors to guide ith robot to avoid colliding with the obstacles. T_i^j is the vector from the ith robot to the jth target, which guides the robot to approach the target when it is too far away and move away from the target when it is too close. O_i^j is the vector from the ith robot to the jth obstacle, which guides that the robot avoids collision with the obstacle. $d_{\min}$ and $d_{\max}$ are the allowed minimum and maximum distances between the swarm robots and targets, respectively.

In summary, the upper layer is the pattern generation layer, which generates an entrapping pattern for swarm robots to follow. The bottom layer is a pattern formation layer; hence, the robots can be guided to appropriate positions to form the entrapping pattern. If the entrapping pattern generated in the upper layer cannot trap the targets, the bottom layer cannot guide the robots to trap the targets. Therefore, this study focuses on the generation layer.

(2) Automated design of the upper layer of the hierarchical GRN model using CMOGP: GRN models designed through experience and intuition not only require a tedious and laborious trail-and-error process, but also cannot guarantee optimal performance for a specific application. Therefore, evolutionary algorithms are proposed as a means to automatically optimize the structures and parameters of GRN models for different application scenarios to generate effective and innovative network structures. To achieve this, CMOGP is used, which combines the constrained multi-objective mechanism (PPS) of PPS-MOEA/D, parameter optimization mechanism of DE, and the multi-criteria decision-making (MCDM) approach. During the evolutionary process, GP is responsible for expressing and evolving GRN models. PPS-MOEA/D is applied to search for the optimal set of design candidates that balance the complexity and performance of the GRN models while satisfying constraints such as avoiding obstacles. DE is used to optimize the parameters of the GRN model, and MCDM is employed to choose the desired GRN model in Pareto-optimal solutions. The pseudo-code of the CMOGP is introduced in Algorithm 1.

In Algorithm 1, initialization is performed in lines 1–3. In line 1, the initial population P_0 is created using the ramped half-and-half method [47]. In line 2, the regulatory parameters of each robot in the initial population are optimized using DE. In line 3, the fitness functions of all the individuals in the initial population are evaluated. In line 6, the offspring population Q_g is generated using crossover and mutation. In lines 7–10, the regulatory parameters of each q in the offspring population are optimized by DE, and the fitness values of q are obtained by evaluating the fitness functions. In line 12, the individuals of new population P_{g+1} are selected through the selection mechanism of the PPS-MOEA/D. The generation counter is updated in line 13. Finally, in line 15, the desired non-dominant and feasible solutions are selected using MCDM.

Algorithm 1: Framework of the CMOGP for entrapping pattern generation

Input:
Ten predefined network motifs;
gen_{max}: the maximum generation.
Output: a set of non-dominant and feasible solutions.
1 Initialize: An initial population P_0 of GRN-based models is randomly created using the ramped half-and-half method;
2 Optimal Parameters: The parameters of these models are optimized by DE;
3 Evaluate: The population P_0 is evaluated using fitness functions;
4 Set $gen = 0$;
5 **while** $gen \leq gen_{max}$ **do**
6 The offspring population Q_g is generated by applying genetic operations;
7 **foreach** $q \in Q_g$ **do**
8 The parameters of q are optimized by DE; // q is an individual in Q_g;
9 Each q in Q_g is evaluated using fitness functions;
10 **end**
11 $R_g \leftarrow P_g \cup Q_g$; // P_g is the gth parent population.
12 The new population P_{g+1} from R_g is formed through the PPS mechanism;
13 $gen = gen + 1$;
14 **end**
15 A desired non-dominant and feasible solution is output by applying a multi-criteria decision-making approach.

(3) GRN model application: To demonstrate the efficacy of the GRN model designed automatically using the CMOGP, the entrapping pattern generated by the GRN model is implemented in several challenging environments. The swarm robots are expected to form entrapping patterns adapted to varying environments and moving targets. The experimental results are discussed in detail in Section 4.

4. Experimental results

Experiments are conducted to verify the effectiveness of the proposed design automation framework. First, the proposed framework generates a GRN model (GRN-1) to entrap two targets moving through two circular obstacles (OBS-1). Next, the proposed framework obtains a new GRN model (GRN-2) to entrap a target moving through a channel containing barbed obstacles (OBS-2). Finally, GRN-1 and GRN-2 are tested, as they migrate to a new environment with a random distribution of multiple circular obstacles (OBS-3). The results of GRN-2 are presented in Appendix A of the supplementary material.

4.1. Entrapping pattern generation in OBS-1

To demonstrate the effectiveness of the design automation framework in OBS-1, the performance of the evolved GRN model is compared with EH-GRN [34] and TH-GRN [35] in OBS-1. For EH-GRN and TH-GRN, both models have fixed structures and apply the covariance matrix adaptation evolution strategy (CMA-ES) [48] to optimize their parameters. In contrast, the proposed framework can generate GRN models with both DE-optimized parameters and a CMOGP-optimized topology. It is worth noting that all three methods are tuned in the same scenario, as shown in Part I of Appendix B. The parameters of the evolutionary process are set as follows:

1. For the proposed framework, the regulatory parameter θ of Positive and Negative are assigned as real numbers ranging from 0 to 1. The regulatory parameter θ of other basic network motif are set as real numbers ranging from 0 to 2. θ of each basic network motif is optimized using DE.
2. For EH-GRN, the regulatory parameter θ_i ($i = 1, 2, \ldots, 14$) of each basic network motif is optimized by CMA-ES. Regulatory parameters θ_7, θ_8, θ_9, θ_{13}, and θ_{14} are assigned real numbers ranging from 0 to 2, and the remaining parameters as real numbers ranging from 0 to 1, according to [34].
3. For TH-GRN, the regulatory parameter θ_i ($i = 1, 2, 3$) of each basic network motif is optimized by CMA-ES. According to [32], regulatory parameters θ_1, θ_2 are assigned real numbers ranging from 0 to 1. θ_3 is set as real numbers ranging from 0 to 2.
4. For the proposed framework, the population size is 50. A GRN hyper-parameter optimization times (ParaNum) is set to 10. Here ParaNum represents the number of times to optimize the parameters when the GRN structure is hold unchanged temporarily.
5. For the EH-GRN and TH-GRN, the population size is set to 200.
6. The crossover and mutation rates of CMOGP and DE are 1.0 and 0.9, and 0.1 and 0.5, respectively.
7. The parameter (ϵ) that control the Push phase to the Pull phase in the PPS mechanism is set to 0.1.
8. The max evaluation number is set to 25000, and each method is run 30 times independently.

In order to better understand the proposed CMOGP in this paper, Table 2 shows the parameter settings of the algorithm.

Before automatically designing the GRN model for OBS-1, some key parameters in the fitness function are set as follows:

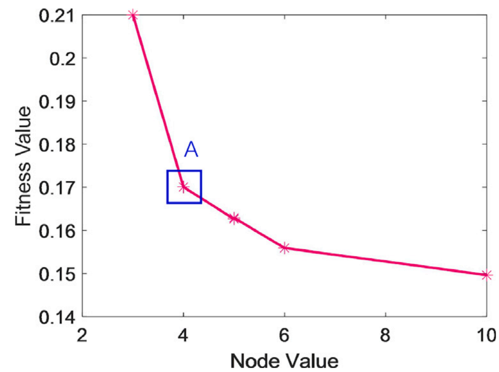


Fig. 4. Non-dominated solutions achieved by CMOGP. Point A is the selected knee point.

Table 2
The parameters are detailed in the proposed CMOGP.

The parameter settings of CMOGP		
Population size		50
A GRN hyper-parameter optimization times		10
Max evaluation number		25000
Control parameter (ϵ)		0.1
Crossover rates	Topology	1
	Parameter	0.1
Mutation rates	Topology	0.9
	Parameter	0.5

(a) Genotype (GP tree) of GRN-1 (b) Phenotype (GRN model) of GRN-1

Fig. 5. Genotype (genetic programming tree) and phenotype (GRN model) of GRN-1. p_1 and p_2 are the positions of targets and obstacles, respectively, to establish corresponding concentration fields. In (a), *ORN* and *Positive* are the two basic network motifs. In (b), G_1 defines two concentration fields regulated by *Positive*. The concentration field M is required to generate the desired entrapping pattern, which is regulated by *ORN*.

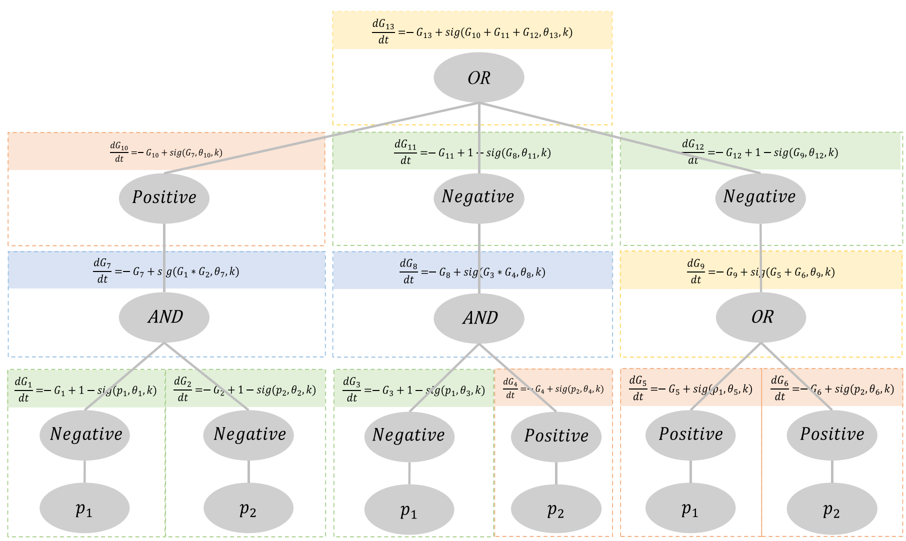


Fig. 6. EH-GRN-1 [34] structure for entrapping pattern generation. The model has a predefined structure. CMA-ES is applied to optimize the regulatory parameters θ_i ($i = 1, \ldots, 14$). p_1 and p_2 are the positions of targets and obstacles respectively to establish corresponding concentration fields.

1. d_{min} and d_{max} are the allowed minimum and maximum distances between the swarm robots and targets, respectively. d_{min} and d_{max} are set to 1 and 2, respectively.
2. $d_{\text{min}}^{\text{obs}}$ is the minimum allowed distance between the swarm robots and obstacles and is set to 1.
3. $d_{\text{min}}^{\text{robot}}$ is the minimum allowed distance between neighboring robots and is set to 1.

All the non-dominated solutions illustrated in Fig. 4 are achieved using CMOGP. As shown in the figure, Point A represents the knee point, and thus the GRN model of Point A, called GRN-1, is selected. Fig. 5(a) and Fig. 5(b) display the genotype (the genetic programming tree of Point A) and the phenotype (the GRN model of Point A), respectively, which is used in the entrapping pattern generation layer of the proposed framework. A discussion of the optimal solutions for the Pareto Front can be found in Appendix B.

The GRN-based distributed controller for entrapping pattern generation developed using GRN-1 is derived as follows:

$$\frac{dT_i}{dt} = \nabla^2 T_i + \gamma_i - T_i \tag{20}$$

$$\frac{dO_i}{dt} = \nabla^2 O_i + \beta_i - O_i \tag{21}$$

$$p_1 = \Sigma_{i=1}^{N_t} T_i \tag{22}$$

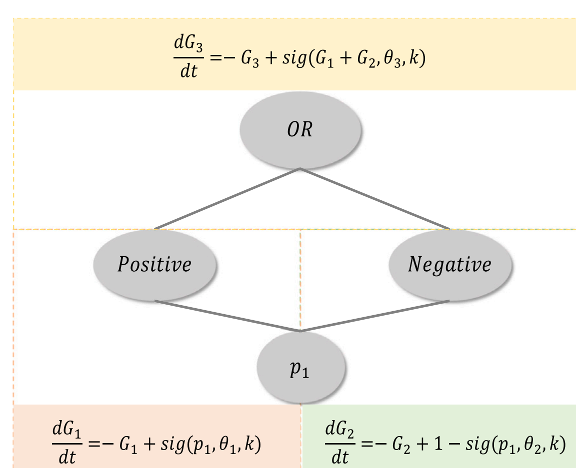


Fig. 7. TH-GRN-1 [35] structure for entrapping pattern generation. The model has a predefined structure. CMA-ES is applied to optimize the regulatory parameters θ_i ($i = 1, 2, 3$). p_1 represents the positions of targets to establish corresponding concentration fields.

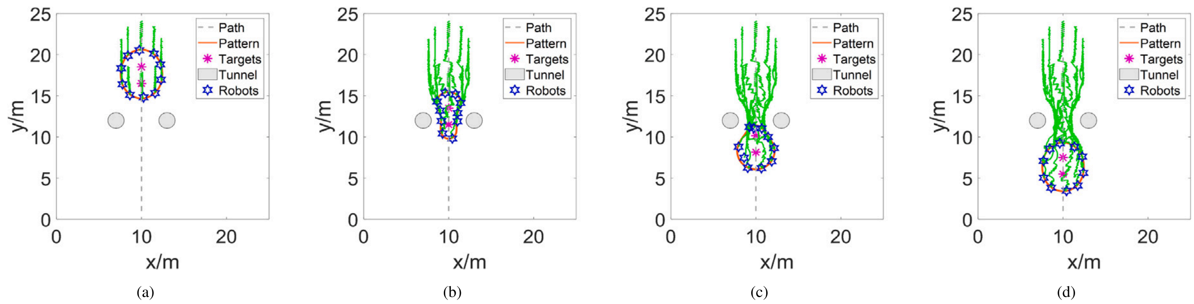


Fig. 8. As the two targets move through the two obstacles, the entrapping pattern generated by the GRN-1 traps the two targets. The obstacles are represented by the two shaded circles. The red contour represents the entrapping pattern. The hexagons represent the swarm robots. The green curves represent the trajectory of the moving swarm robots. The gray dotted line represents the trajectory of the moving targets. (a)–(d) illustrate several typical scenarios when the two targets are in different positions and the entrapping pattern that helps the swarm robots trap the targets. The robots follow the pattern and trap the two targets.

$$p_2 = \Sigma_{i=1}^{N_o} O_i \tag{23}$$

$$\frac{dG_1}{dt} = -G_1 + sig(p_2, \theta_1, k) \tag{24}$$

$$\frac{dM}{dt} = -M + sig(G_1 + (1 - p_1), \theta_2, k) \tag{25}$$

where γ_i and β_i represent the positions of the ith target and ith obstacle, respectively. T_i and O_i represent the concentration field components of γ_i and β_i formed by the ith target and ith obstacle, respectively. p_1 and p_2 represent the concentration fields produced by all detected targets and obstacles, respectively. N_t and N_o are the total numbers of targets and obstacles, respectively. ∇^2 is a Laplace operator, which is defined as the second derivative of T_i and O_i. G_1 fuses the concentration field from p_2, which is regulated by the *Positive* regulation (as listed in Table 1). M fuses the concentration field from G_1 and p_1, which is regulated by the *ORN* regulation, to generate entrapping patterns. By Eqs. (24) and (25), some closed contour lines of concentration values are obtained in the M concentration field. The concentration contours around the target are used as the candidate entrapping patterns. In addition, θ_1 and θ_2 are 0.702 and 1.189, respectively, in Eqs. (24) and (25).

For the generated entrapping pattern to be used by the lower layer of the GRN, the contour of the pattern must first be extracted [49]. A concentration contour is selected as the entrapping pattern according to the following conditions: (1) The minimum distance between the contour line and the target/obstacle should exceed the predefined safe distance, to avoid collision. (2) The farthest distance between the contour line and the target should be less than the allowed maximum distance between the robot and target, which can ensure entrapping the target. (3) The contour line is centered on the target as much as possible, which ensures that the target is encircled. With these selection criteria, a contour line is selected and considered as an entrapping pattern that is used by the lower layer of the GRN model to guide the movements of the swarm robots.

Fig. 6 illustrates a EH-GRN model, namely EH-GRN-1. In this model, all regulatory parameters θ_i ($i = 1, \dots, 14$) are optimized by CMA-ES: $\theta_1 = 0.6958$, $\theta_2 = 1$, $\theta_3 = 0.5758$, $\theta_4 = 0.1183$, $\theta_5 = 0.0432$, $\theta_6 = 0$, $\theta_7 = 0.4614$, $\theta_8 = 0$, $\theta_9 = 0.8710$, $\theta_{10} = 0.3903$, $\theta_{11} = 0.5621$, $\theta_{12} = 0.7134$, $\theta_{13} = 0.8934$. In addition, Fig. 7 illustrates a TH-GRN model. In this model, all regulatory parameters θ_i ($i = 1, 2, 3$) are optimized by CMA-ES: $\theta_1 = 0.5004$, $\theta_2 = 0.0822$, $\theta_3 = 1.9732$.

In Fig. 8, GRN-1 generates an adaptive entrapping pattern to entrap two moving targets as they navigate through two circular obstacles. For example, as the two targets approach the obstacles, GRN-1 generates an ellipsoidal pattern, which is highlighted by the red line in Fig. 8(a), to lead the swarm robots to entrap the two targets. When the targets are in the middle of the circular obstacles, as shown in Fig. 8(b), the entrapping pattern changes from an ellipsoidal to dumbbell-like pattern, which leads the swarm robots to entrap the two targets without colliding with the obstacles. This pattern change may have occurred because the concentration field formed by the two circular obstacles is compressed, and thus changes the shape of the concentration field formed by the targets. As the two targets are farther from the two circular obstacles, the entrapping pattern reverts from droplet-like to ellipsoidal, as shown in Fig. 8(c)–(d).

As shown in Fig. 9, the EH-GRN-1 generates an entrapping pattern to entrap the two targets as they pass through two circular obstacles. When the two targets approach the circular obstacles, the entrapping pattern generated by the model not only entraps the two targets but also the two obstacles, as shown in Fig. 9(a). This type of entrapping pattern is generated because the concentration field formed by the two circular obstacles is fused with that formed by the two targets. Although swarm robots guided by the pattern entrap the two targets, they are not evenly distributed in the pattern. When the two targets are in the middle of two circular obstacles, an ellipsoidal pattern is generated, as shown in Fig. 9(b). However, the robots are clustered on one side of the pattern. When the two targets are farther from the two circular obstacles, the generated pattern still encircles both targets and obstacles. However, some robots of the swarm remain concentrated on one side of the obstacles and do not follow and trap the targets, as shown in Fig. 9(c)–(d). Although the remaining robots achieve the entrapping task temporarily, if the targets keep moving in a similar environment, the number of entrapping robots may gradually decrease and the entrapping task may fail after all. To verify this conjecture, this study adds a set of experiments to demonstrate what will happen if the targets pass through two groups of circular obstacles, which is shown in the fourth part of Appendix B.

For TH-GRN, Fig. 10 shows the process of the entrapping patterns generated by TH-GRN to guide the swarm robots to entrap the targets in OBS-1. When the targets pass through the obstacles, the patterns generated by TH-GRN traverse, as shown in Fig. 10(b). This is because the upper layer of TH-GRN only considers the targets' location information to generate the entrapping pattern, which ignores the influence of the obstacles. On the other hand, the inter-layer of TH-GRN utilizes the obstacle information to guide the robots to avoid obstacles. Fig. 10(c)–(d) show that the robots can move well to the entrapping pattern. As a result, the entrapping pattern generated by TH-GRN is always a circular-like shape with no shape change. It is worth noting that when the environment is complex, this unchanged shape may cause many robots to be blocked by obstacles, and thus cannot consistently entrap the targets, as shown in Fig. 15.

To quantify and compare the performances of the EH-GRN-1, TH-GRN and GRN-1 with OBS-1, Fig. 11(a)–(b) present the curves of C_e and D_v evolving over time. In Fig. 11(a)–(b), when the targets do not pass through two obstacles, the C_e and D_v of GRN-1 converges to a near-zero value, which indicates that the swarm robots basically reach and distribute well in the entrapping pattern. When the targets pass through the two obstacles, C_e and D_v of GRN-1 fluctuate significantly during time steps 170–250. This is because the entrapping pattern generated by GRN-1 changes drastically during this period, to adapt

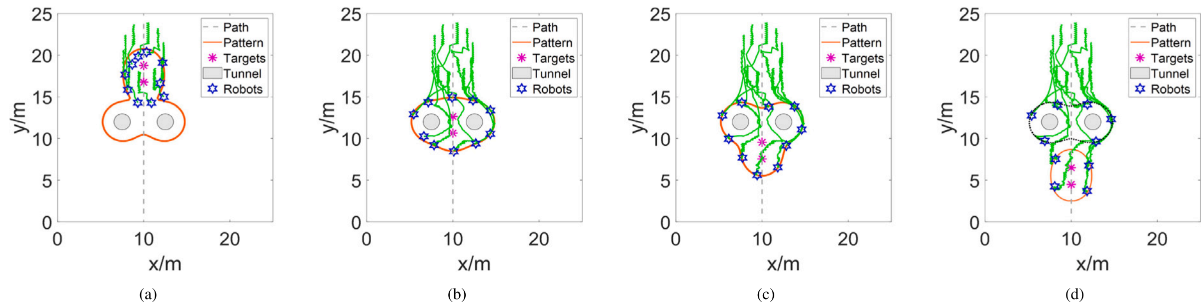


Fig. 9. Schematic illustration of the trapping pattern generated by EH-GRN-1 [34] when two targets pass through two obstacles. The obstacles are represented by the two shaded circles. The red contour represents the entrapping pattern. Hexagonal shapes represent the swarm robots. The green curves represent the trajectory of the moving swarm robots. The gray dotted line represents the trajectory of the moving targets. (a)–(d) illustrate several typical scenarios for the two targets in different positions, and the entrapping patterns that lead the swarm robots to trap the targets. The swarm robots cannot follow the targets and fail to trap them.

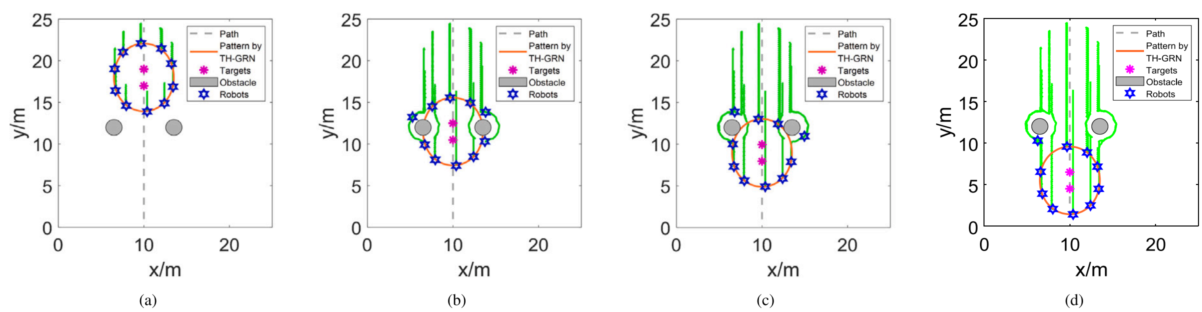


Fig. 10. As the two targets move through the two obstacles, the entrapping pattern generated by TH-GRN [35] entraps the two targets. (a)–(d) illustrate several typical scenarios when the two targets are in different positions, and the entrapping pattern that helps the swarm robots entrap the targets.

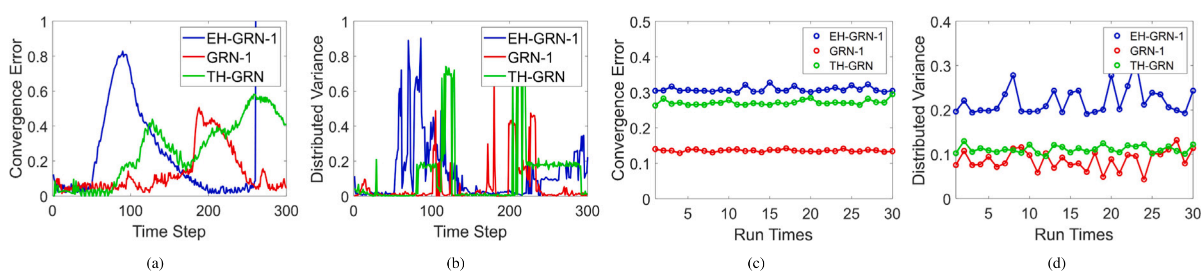


Fig. 11. The statistical values of the two evaluations on OBS-1 achieved by EH-GRN-1 [34], TH-GRN [35] and GRN-1. (a) Curve of C_e evolving over time. (b) Curve of D_v evolving over time. (c) Curve of C_e in 30 independent runs. (d) Curve of D_v in 30 independent runs.

to the environmental variation caused by the presence of obstacles. It will take some time for the swarm robots to follow up and evenly distribute themselves in the pattern again, as shown in Fig. 8(b)–(c). For EH-GRN-1, C_e and D_v fluctuate significantly during the time steps 70–130. This is because when the targets are close to the obstacles, the entrapping pattern generated by EH-GRN-1 suddenly changes from entrapping the targets to entrapping both the targets and obstacles, causing the entrapping pattern to become larger, as shown in Fig. 9(a). In addition, after time step 265, C_e suddenly increases, as shown in Fig. 11(a). This is because when the two targets escape far away from the two circular obstacles, some robots of the swarm remain blocked by the obstacles and cannot follow the targets, as shown in Fig. 9(c)–(d). Fig. 11(c)–(d) shows the statistical values of the two evaluations on OBS-1 achieved by EH-GRN-1, TH-GRN and GRN-1 in 30 independent runs. We can clearly see that the statistical values of C_e and D_v for GRN-1 are much smaller than those for EH-GRN-1 and TH-GRN for each number of runs. Table 3 presents the average values of C_e and D_v over 30 independent runs, which demonstrate that the performance of GRN-1 is better than that of EH-GRN-1 and TH-GRN in OBS-1. It is worth noting that the values of C_e and D_v for each independent run are shown in Part V of Appendix B.

4.2. Transferring the obtained models to OBS-3

To demonstrate the transferability of the GRN model generated by the proposed framework, GRN-1 is migrated to a new application scenario with a large number of randomly distributed obstacles. For comparison, EH-GRN-1 and TH-GRN are also tested with OBS-3.

When the GRN model generated by the proposed framework is migrated to the new application scenario, an interesting phenomenon emerges, as shown in Fig. 12. According to the change of the surrounding environment of the escaping target, the entrapping pattern generated by the proposed method can be adapted to very irregular shapes, as shown in Fig. 12, which leads the swarm robots to avoid the obstacles and entrap the target. This kind of behavior resembles a phenomenon that can be observed in the biological world. For example, white blood cells (phagocytizes) change their shape to shuttle between crowded cells, and engulf a bacterium (as shown in Fig. 13).[1] In Fig. 14, we clearly observe the merging of the entrapping pattern generated by EH-GRN-1 with the obstacles around the target to entrap the target.

[1] https://embryology.med.unsw.edu.au/embryology/index.php/Movie_-_Neutrophil_chasing_bacteria

Table 3
Mean and standard deviation values of Convergence Error and Distributed Variance, obtained by EH-GRN-1 (EH) [34], TH-GRN (TH) [35] and GRN-1 on OBS-1 and OBS-3.

	Instance		GRN-1	EH [34]	TH [35]	t-test (p-value)	
						GRN-1 vs EH	GRN-1 vs TH
Convergence error	OBS-1	mean	**1.3581E−01**	3.0819E−01	2.7143E−01	6.35E−41	7.33E−37
		std	3.1688E−03	7.2254E−03	7.1285E−03		
	OBS-3	mean	**8.7313E−02**	2.2014E−01	1.1120E−01	1.35E−18	8.34E−25
		std	2.1643E−02	3.0694E−02	8.4375E−03		
	Instance		GRN-1	EH [34]	TH [35]	t-test (p-value)	
						GRN-1 vs EH	GRN-1 vs TH
Distributed variance	OBS-1	mean	**8.7313E−02**	2.2014E−01	1.1120E−01	1.61E−19	4.25E−06
		std	2.1643E−02	3.0694E−02	8.4375E−03		
	OBS-3	mean	**3.4045E−02**	4.0491E−01	3.3760E−01	1.09E−36	3.11E−27
		std	4.2570E−03	2.3030E−02	4.0088E−02		

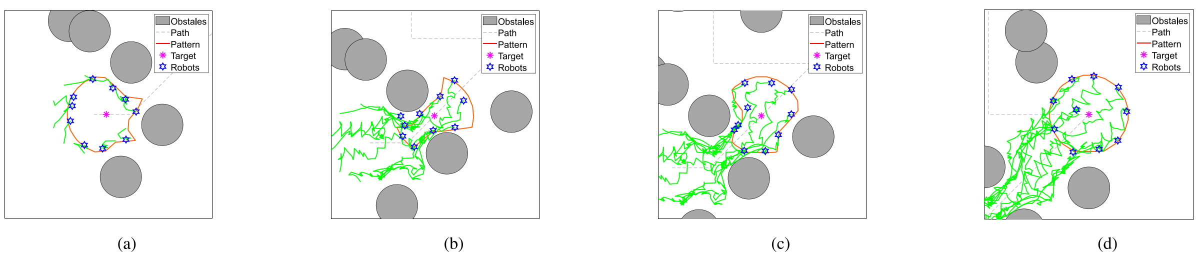


Fig. 12. When a target passes through OBS-3, with a large number of randomly distributed circular obstacles, the pattern generated by GRN-1 entraps the target. The green curves represent the trajectory of the moving swarm robots. The gray dotted line represents the trajectory of the moving targets. (a)–(d) illustrate the four typical scenarios in which the entrapping pattern leads the swarm robots to trap the target as it moves to different locations of OBS-3. The robots firmly trap the targets in all cases.

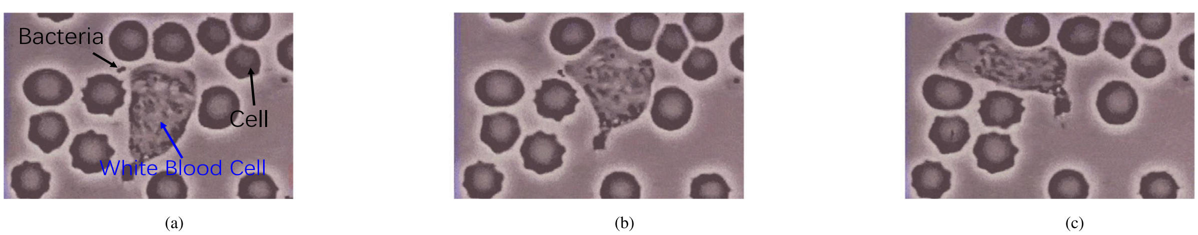


Fig. 13. A white blood cell phagocytizes a bacterium throughout the process.

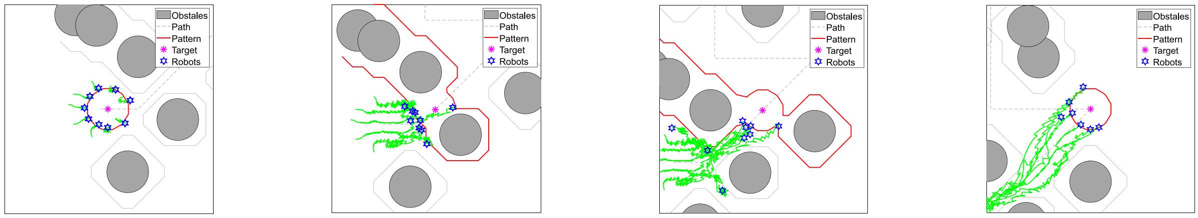
However, the entrapping pattern fails to lead the swarm robots to entrap the target as the target escapes, possibly because the merging of the pattern produces a new and enlarged pattern that entrapped the target and obstacles simultaneously, as shown in Fig. 14(b). Because the new entrapping pattern is largely distracted by the obstacles and does not focus on the target, it fails to lead all swarm robots to form a pattern to firmly entrap the target. For TH-GRN, when the target passes between obstacles, quite some robots are blocked by obstacles and thus cannot entrap the targets well, as shown in Fig. 15(b)–(c). This is because the entrapping pattern generated by TH-GRN cannot adaptively change its shape according to the environment, resulting in many robots being blocked by the obstacles.

To quantify and compare the performances of GRN-1, EH-GRN-1 and TH-GRN in the OBS-3, Fig. 16(a)–(b) present the curves of C_e and D_v evolving over time. In Fig. 16(a), GRN-1 obtained through the proposed framework performs significantly better than EH-GRN-1 and TH-GRN. This is because the entrapping pattern generated by EH-GRN-1 entraps the target and obstacles simultaneously, producing a new and enlarged pattern. If the obstacle size (such as a combination of multiple obstacles) is large, the entrapping pattern generated by EH-GRN-1 will be large too, as shown in Fig. 14(b)–(c). In addition, the entrapping pattern generated by TH-GRN is always a circular-like shape, which prevents it from adapting itself to variable shapes to accommodate with changing obstacle environment, as shown in Fig. 15(b)–(c). Therefore, swarm robots following patterns generated by EH-GRN-1 and TH-GRN do not have competitive convergence and distribution as those of GRN-1, as shown in Fig. 16. By comparing the results shown in Fig. 16(a)–(b), we clearly find that the value of C_e and D_v of GRN-1 are significantly smaller than those of EH-GRN-1 and TH-GRN. In addition, Fig. 16(c)–(d) illustrate the values of the two evaluations on OBS-3 achieved by GRN-1, EH-GRN-1 and TH-GRN in 30 independent runs. We can clearly see that the values of C_e and D_v for GRN-1 are much smaller than those of EH-GRN-1 and TH-GRN in each run. Statistical significance is also verified by p-value, as shown in Table 3. Therefore, it is considered that the performance of GRN-1 is better than that of EH-GRN-1 and TH-GRN in OBS-3, and that GRN-1 can be successfully transferred to OBS-3. It is worth noting that the values of C_e and D_v for each independent run are shown in Part V of Appendix B.

4.3. Experiments using e-puck robots

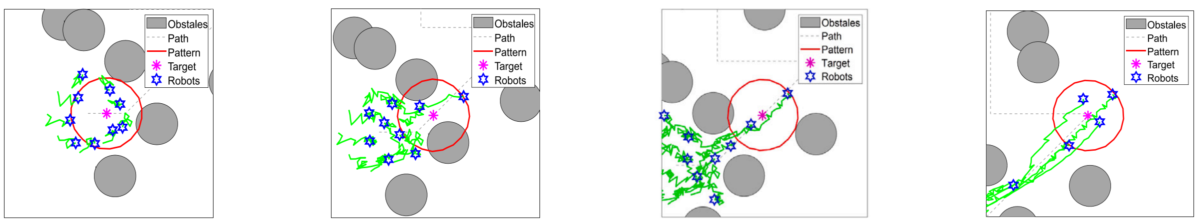
4.3.1. Entrapping multiple targets simultaneously

The proposed method has the capability to generate patterns that can entrap multiple targets, which actually stems from the ability of the proposed method to generate adaptable entrapping formation without predefined pattern. For example, when it is necessary that the existing large pattern to encircle some clustered targets needs to divide itself into multiple small ones to entrap the scattered targets when the escape in different directions, it will do so “automatically” without any



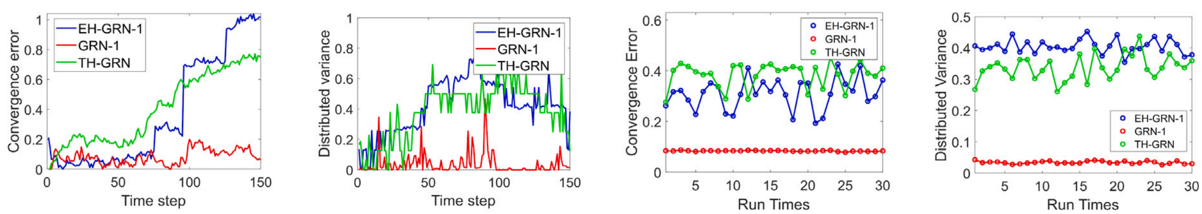
(a) (b) (c) (d)

Fig. 14. As a target passes through OBS-3, with numerous randomly distributed circular obstacles, the entrapping pattern generated by EH-GRN-1 [34] entraps the target. The green curves represent the trajectory of the moving swarm robots. The gray dotted line represents the trajectory of the moving targets. (a)–(d) illustrate the four typical scenarios in which the entrapping pattern leads the swarm robots to trap the target in OBS-3. The pattern merges with the boundary of the obstacles, and the swarm robots are unevenly distributed in a significantly enlarged pattern contour. Consequently, many robots deviate from the pattern and fail to trap the target as it moves to the location of (d).



(a) (b) (c) (d)

Fig. 15. When a target passes through OBS-3, with a large number of randomly distributed circular obstacles, the pattern generated by TH-GRN [35] entraps the target. (a)–(d) illustrate the four typical scenarios in which the entrapping pattern leads the swarm robots to trap the target as it moves to different locations of OBS-3.



(a) (b) (c) (d)

Fig. 16. The statistical values of the two evaluations on OBS-3 achieved by GRN-1, EH-GRN-1 [34] and TH-GRN [35]. (a) Curve of C_e evolving over time. (b) Curve of D_v evolving over time. (c) Curve of C_e in 30 independent runs. (d) Curve of D_v in 30 independent runs.

specifications. Fig. 17 shows a physical experiment conducted using E-Puck robot systems. The two targets (in blue lights) are first surrounded by two swarms of robots (in red lights) in two circles. Then the two targets begin to escape and approach each other. When they almost move into the same position, the two circles gradually merge into a big one, entrapping the two targets as a whole. After that, the two targets continue to escape in opposite directions. When they get apart enough distance, the big circle is divided into two small ones again, with each entrapping one target separately. During the whole process, no specific instructions are given to the swarm robots to make group merge and/or division, which resembles the process of cell fusion and cell division in biology. The above experiment video is given in Appendix C.

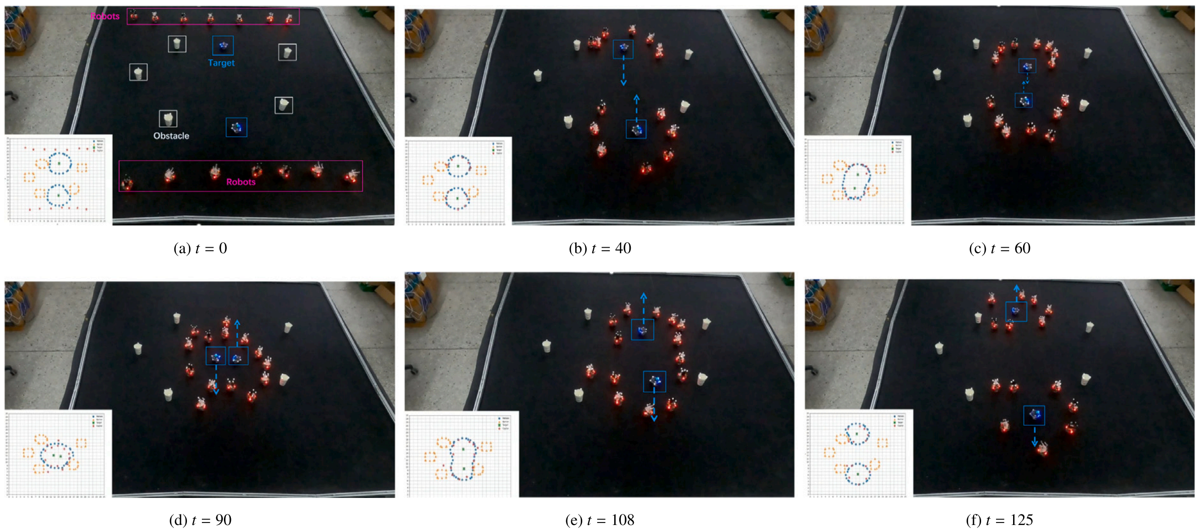
4.3.2. Entrapping targets when some robots fail to function

Because the proposed method follows a fully distributed control scheme, if some robots in the swam fail to function, the remaining robots can still fulfill the task of entrapment. As shown in Fig. 18, from the beginning, the first three snapshots Fig. 18(a)–(b) show that the surrounding swarm robots (in red) entrap the target (in blue) successfully. Then the target tries to escape through the obstacles (in white). The swarm robots follow the target and keep the entrapment formation in Fig. 18(c). Then two robots in the swarm fail to function suddenly, and stand still (in yellow), as shown in Fig. 18(d). However, the remaining swarm robots still bypass the obstacles and the failed robots as well. The swarm robots finally entrap the target tightly, as demonstrated in Fig. 18(e)–(f). The video of the physical experiment can also be found in Appendix C.

5. Conclusion

The design of GRN models for the self-organization of swarm robots is usually a tedious and laborious trial-and-error process that is largely dependent on the experience and intuition of human experts. Furthermore, a GRN model specially designed for a particular application scenario cannot guarantee its applicability in a new scenario. To address these issues, this study proposes a design automation framework to generate entrapping patterns using CMOGP, a program that synthesizes GRN models by reconfiguring a predefined set of basic GRN network motifs. By using the proposed framework, a comprehensive investigation of the evolved candidates in the Pareto front provides knowledge for designing a GRN model that can generate patterns to perform predefined tasks. The experimental results show that the evolved GRN models using this framework can generate patterns that guide swarm robots to successfully entrap targets in a variety of challenging scenarios.

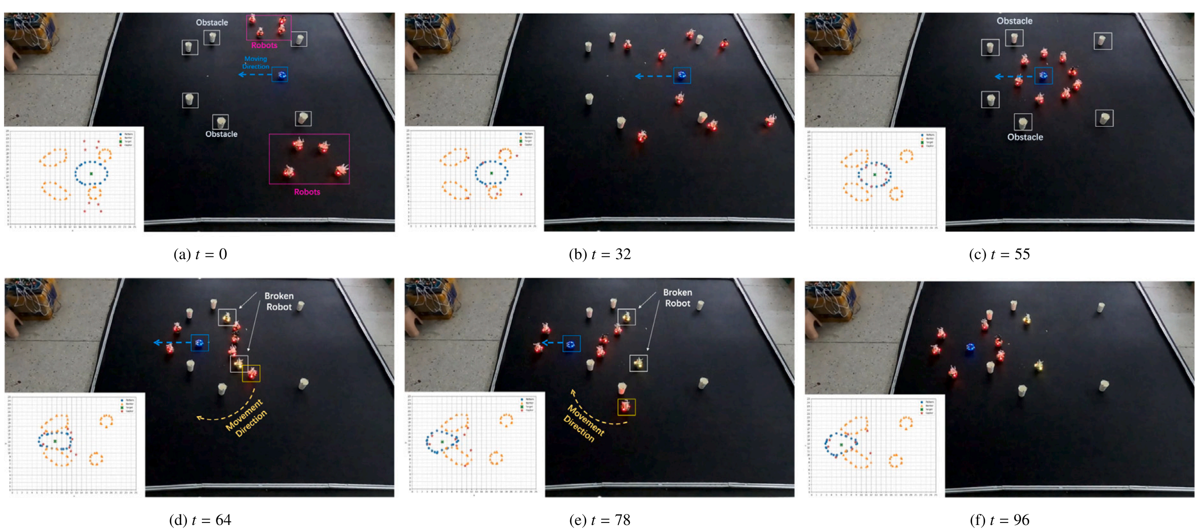
The evolved GRN models have three main features: (1) According to the change of surrounding environment of the escaping targets, the entrapping patterns generated by the evolved GRN models can be adapted to very irregular shapes to avoid obstacles while entrapping the targets. (2) The entrapping patterns generated by the evolved GRN models can achieve group division and merge to enable entrapping multiple targets at the same time, without any predefined specifications. (3) The proposed method follows a fully distributed control scheme, meaning that even if some robots of the swarm fail to function, the remaining robots can still fulfill the entrapping task.



(a) $t = 0$ (b) $t = 40$ (c) $t = 60$

(d) $t = 90$ (e) $t = 108$ (f) $t = 125$

Fig. 17. The entrapping pattern generated by our proposed method entraps two targets in an environment with obstacles. (a) The two targets (in blue light) are first surrounded by two swarms of robots (in red light) in two circles. (b)–(c) Then the two targets begin to escape and approach each other. (d) When they almost move into the same position, the two circles gradually merge into a big one, entrapping the two targets as a whole. (e) After that, the two targets continue to escape in opposite directions. (f) When they get apart enough distance, the big circle is divided into two small ones again, with each entrapping one target separately.



(a) $t = 0$ (b) $t = 32$ (c) $t = 55$

(d) $t = 64$ (e) $t = 78$ (f) $t = 96$

Fig. 18. Entrapping pattern generated by our proposed method entraps one target in an environment with obstacles. The target is the robot with blue light. Normally working robots are lit up in red, and failed robots are shown in yellow. The stationary obstacles have white color. (a)–(b) the surrounding swarm robots (in red) entrap the target (in blue) successfully. (c) the target tries to escape through the obstacles (in white). The swarm robots follow the target and keep the entrapment formation. (d) two robots in the swarm fail to function suddenly, and stand still (in yellow). (e)–(f) the remaining swarm robots still bypass the obstacles and the failed robots as well, and entrap the target tightly.

CRediT authorship contribution statement

Zhun Fan: Conceptualization, Methodology. **Zhaojun Wang:** Software, Investigation, Writing – original draft, Writing – review & editing. **Wenji Li:** Validation, Writing – review & editing. **Xiaomin Zhu:** Methodology. **Bingliang Hu:** Methodology. **An-Min Zou:** Formal analysis. **Weidong Bao:** Formal analysis. **Minqiang Gu:** Formal analysis. **Zhifeng Hao:** Conceptualization, Methodology. **Yaochu Jin:** Methodology, Visualization.

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.

Acknowledgments

This research was supported in part by National Key R&D Program of China (grant number 2021ZD0111501, 2021ZD0111502), the Key Laboratory of Digital Signal and Image Processing of Guangdong Province, the Key Laboratory of Intelligent Manufacturing Technology (Shantou University), Ministry of Education, the Science and Technology Planning Project of Guangdong Province of China (grant number 180917144960530), the Project of Educational Commission of Guangdong Province of China (grant number 2017KZDXM032), the State Key Lab of Digital Manufacturing Equipment & Technology, China (grant number DMETKF2019020), National Natural Science Foundation of

China (grant number 62176147), STU Scientific Research Foundation for Talents (grant number NTF21001), Science and Technology Planning Project of Guangdong Province of China (grant number 2019A050520001, 2021A0505030072, 2022A1515110660, 2022A1515110566, 2023A1515011574), Science and Technology Special Funds Project of Guangdong Province of China (grant number STKJ2021176, STKJ2021019), and Guangdong Special Support Program for Outstanding Talents (2021JC06X549).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.swevo.2023.101337.

References

[1] A. Scheidler, A. Brutschy, E. Ferrante, M. Dorigo, The k-unanimity rule for self-organized decision-making in swarms of robots, IEEE Trans. Cybern. 46 (5) (2015) 1175–1188.
[2] X. Yi, A. Zhu, S.X. Yang, C. Luo, A bio-inspired approach to task assignment of swarm robots in 3-D dynamic environments, IEEE Trans. Cybern. 47 (4) (2016) 974–983.
[3] A. Khan, B. Rinner, A. Cavallaro, Cooperative robots to observe moving targets, IEEE Trans. Cybern. 48 (1) (2016) 187–198.
[4] A. Murai, H. Matsukura, R. Takemura, H. Ishida, Active airflow generation to assist robotic gas source localization: Initial experiments in outdoor environment, ECS Trans. 75 (16) (2016) 65–72.
[5] A. Macwan, J. Vilela, G. Nejat, B. Benhabib, A multirobot path-planning strategy for autonomous wilderness search and rescue, IEEE Trans. Cybern. 45 (9) (2014) 1784–1797.
[6] W. Zhao, Q. Meng, P.W. Chung, A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario, IEEE Trans. Cybern. 46 (4) (2015) 902–915.
[7] S. Kim, H. Oh, J. Suk, A. Tsourdos, Coordinated trajectory planning for efficient communication relay using multiple UAVs, Control Eng. Pract. 29 (2014) 42–49.
[8] B. Fu, Y. Xiao, X. Liang, C.P. Chen, Bio-inspired group modeling and analysis for intruder detection in mobile sensor/robotic networks, IEEE Trans. Cybern. 45 (1) (2014) 103–115.
[9] L.E. Barnes, M.A. Fields, K.P. Valavanis, Swarm formation control utilizing elliptical surfaces and limiting functions, IEEE Trans. Syst. Man Cybern. B 39 (6) (2009) 1434–1445.
[10] M. Rubenstein, A. Cornejo, R. Nagpal, Programmable self-assembly in a thousand-robot swarm, Science 345 (6198) (2014) 795–799.
[11] Z. Ji, H. Yu, A new perspective to graphical characterization of multiagent controllability, IEEE Trans. Cybern. 47 (6) (2016) 1471–1483.
[12] Y. Zhao, B. Li, J. Qin, H. Gao, H.R. Karimi, H_∞ Consensus and synchronization of nonlinear systems based on a novel fuzzy model, IEEE Trans. Cybern. 43 (6) (2013) 2157–2169.
[13] L. Han, X. Dong, Q. Li, Z. Ren, Formation tracking control for time-delayed multi-agent systems with second-order dynamics, Chin. J. Aeronaut. 30 (1) (2017) 348–357.
[14] X. Yu, J. Ma, N. Ding, A. Zhang, Cooperative target enclosing control of multiple mobile robots subject to input disturbances, IEEE Trans. Syst. Man Cybern. Syst. 51 (6) (2021) 3440–3449.
[15] Z. Yang, C. Chen, S. Zhu, X.-P. Guan, G. Feng, Distributed entrapping control of multi-agent systems using bearing measurements, IEEE Trans. Automat. Control 66 (12) (2020) 5696–5710.
[16] H. Kawakami, T. Namerikawa, Virtual structure based target-enclosing strategies for nonholonomic agents, in: 2008 IEEE International Conference on Control Applications, IEEE, 2008, pp. 1043–1048.
[17] K. Sato, N. Maeda, Target-enclosing strategies for multi-agent using adaptive control strategy, in: 2010 IEEE International Conference on Control Applications, IEEE, 2010, pp. 1761–1766.
[18] H. Rezaee, F. Abdollahi, A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots, IEEE Trans. Ind. Electron. 61 (1) (2013) 347–354.
[19] G. Antonelli, F. Arrichiello, S. Chiaverini, The NSB control: a behavior-based approach for multi-robot systems, Paladyn, J. Behav. Robot. 1 (1) (2010) 48–56.
[20] N. Phung, M. Kubo, H. Sato, S. Iwanaga, Agreement algorithm using the trial and error method at the macrolevel, Artif. Life Robot. 23 (4) (2018) 564–570.
[21] J. Ma, H. Lu, J. Xiao, Z. Zeng, Z. Zheng, Multi-robot target encirclement control with collision avoidance via deep reinforcement learning, J. Intell. Robot. Syst. 99 (2) (2020) 371–386.
[22] Y. Wang, L. Dong, C. Sun, Cooperative control for multi-player pursuit-evasion games with reinforcement learning, Neurocomputing 412 (2020) 101–114.
[23] J. Park, J. Lee, T. Kim, I. Ahn, J. Park, Co-evolution of predator-prey ecosystems by reinforcement learning agents, Entropy 23 (4) (2021) 461.
[24] A. Jahn, R.J. Alitappeh, D. Saldaña, L.C. Pimenta, A.G. Santos, M.F. Campos, Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 273–278.
[25] S. Zhang, M. Liu, X. Lei, Y. Huang, F. Zhang, Multi-target trapping with swarm robots based on pattern formation, Robot. Auton. Syst. 106 (2018) 1–13.
[26] D. Li, S.S. Ge, W. He, C. Li, G. Ma, Distributed formation control of multiple Euler-Lagrange systems: A multilayer framework, IEEE Trans. Cybern. 52 (5) (2020) 3325–3332.
[27] H. Oh, A.R. Shirazi, C. Sun, Y. Jin, Bio-inspired self-organising multi-robot pattern formation: A review, Robot. Auton. Syst. 91 (2017) 83–100.
[28] R. Doursat, H. Sayama, O. Michel, A review of morphogenetic engineering, Nat. Comput. 12 (4) (2013) 517–535.
[29] S. Kondo, T. Miura, Reaction-diffusion model as a framework for understanding biological pattern formation, Science 329 (5999) (2010) 1616–1620.
[30] L. Bai, M. Eyiyurekli, P.I. Lelkes, D.E. Breen, Self-organized sorting of heterotypic agents via a chemotaxis paradigm, Sci. Comput. Program. 78 (5) (2013) 594–611.
[31] A.R. Shirazi, Bio-Inspired Self-Organizing Swarm Robotics, University of Surrey (United Kingdom), 2017.
[32] Y. Jin, H. Guo, Y. Meng, A hierarchical gene regulatory network for adaptive multirobot pattern formation, IEEE Trans. Syst. Man Cybern. B 42 (3) (2012) 805–816.
[33] X. Peng, S. Zhang, X. Lei, Multi-target trapping in constrained environments using gene regulatory network-based pattern formation, Int. J. Adv. Robot. Syst. 13 (5) (2016) 1729881416670152.
[34] H. Oh, Y. Jin, Evolving hierarchical gene regulatory networks for morphogenetic pattern formation of swarm robots, in: 2014 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2014, pp. 776–783.
[35] Y. Yuan, Z. Fan, X. Zhu, M. Wu, L. Ma, T. Fang, Z. Wang, W. Bao, Y. Zhou, H. Chen, et al., TH-GRN model based collective tracking in confined environment, in: International Conference on Swarm Intelligence, Springer, 2019, pp. 33–43.
[36] Y. Jin, Y. Meng, Emergence of robust regulatory motifs from in silico evolution of sustained oscillation, BioSystems 103 (1) (2011) 38–44.
[37] S.A. Thomas, Y. Jin, Evolving connectivity between genetic oscillators and switches using evolutionary algorithms, J. Bioinform. Comput. Biol. 11 (03) (2013) 1341001.
[38] S.A. Thomas, Y. Jin, E. Laing, C.P. Smith, Reconstructing regulatory networks in streptomyces using evolutionary algorithms, in: 2013 13th UK Workshop on Computational Intelligence, UKCI, IEEE, 2013, pp. 24–30.
[39] Z. Fan, K. Seo, J. Hu, E.D. Goodman, R.C. Rosenberg, A novel evolutionary engineering design approach for mixed-domain systems, Eng. Optim. 36 (2) (2004) 127–147.
[40] J.-F. Dupuis, Z. Fan, E.D. Goodman, Evolutionary design of both topologies and parameters of a hybrid dynamical system, IEEE Trans. Evol. Comput. 16 (3) (2011) 391–405.
[41] J.-F. Dupuis, Z. Fan, E. Goodman, Evolutionary design of discrete controllers for hybrid mechatronic systems, Internat. J. Systems Sci. 46 (2) (2015) 303–316.
[42] E. Osaba, E. Villar-Rodriguez, J. Del Ser, A.J. Nebro, D. Molina, A. LaTorre, P.N. Suganthan, C.A.C. Coello, F. Herrera, A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems, Swarm Evol. Comput. 64 (2021) 100888.
[43] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman, Push and pull search for solving constrained multi-objective optimization problems, Swarm Evol. Comput. 44 (2019) 665–679.
[44] R. Storn, K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. (ISSN: 1573-2916) 11 (4) (1997) 341–359.
[45] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, Network motifs: simple building blocks of complex networks, Science 298 (5594) (2002) 824–827.
[46] P.M. Bowers, S.J. Cokus, D. Eisenberg, T.O. Yeates, Use of logic relationships to decipher protein network organization, Science 306 (5705) (2004) 2246–2249.
[47] W.B. Langdon, Size fair and homologous tree genetic programming crossovers, Genet. Program. Evol. Mach. 1 (1/2) (2000) 95–119.
[48] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.
[49] H. Oh, Y. Jin, Adaptive swarm robot region coverage using gene regulatory networks, in: Conference Towards Autonomous Robotic Systems, Springer, 2014, pp. 197–208.