A Cellular Mechanism for Multi-Robot Construction via Evolutionary Multi-Objective Optimization of A Gene Regulatory Network

Honglian Guo, Yan Meng, Yaochu Jin

2009

Preprint:

This is an accepted article published in BioSystems. The final authenticated version is available online at: https://doi.org/[DOI not available]

To appear in BioSystems

A Cellular Mechanism for Multi-Robot Construction via Evolutionary Multi-Objective Optimization of A Gene Regulatory Network

Hongliang Guo[a], Yan Meng[a*], and Yaochu Jin[b]

[a]*Department of Electrical and Computer Engineering, Stevens Institute of Technology, NJ 07030, USA*
hguo@stevens.edu, yan.meng@stevens.edu
[b]*Honda Research Institute Europe, Carl-Legien-Str. 30, 3073 Offenbach, Germany*
yaochu.jin@honda-ri.de

** Corresponding author: email: yan.meng@stevens.edu, phone: 1-201-216-5496, fax: 1-201-216-8246*

Abstract

A major research challenge of multi-robot systems is to predict the emerging behaviors from the local interactions of the individual agents. Biological systems can generate robust and complex behaviors through relatively simple local interactions in a world characterized by rapid changes, high uncertainty, infinite richness, and limited availability of information. Gene Regulatory Networks (GRNs) play a central role in understanding natural evolution and development of biological organisms from cells. In this paper, inspired by biological organisms, we propose a distributed GRN-based algorithm for a multi-robot construction task. Through this algorithm, multiple robots can self-organize autonomously into different predefined shapes, and self-reorganize adaptively under dynamic environments. This developmental process is evolved using a multi-objective optimization algorithm to achieve a shorter travel distance and less convergence time. Furthermore, a theoretical proof of the system's convergence is also provided. Various case studies have been conducted in the simulation, and the results show the efficiency and convergence of the proposed method.

Keywords: Gene regulatory networks; Multi-cellular systems; Distributed multi-robot systems; Self-organizing systems; Multi-objective optimization.

I.

1. Introduction

A major research challenge of self-organizing collective systems, such as multi-robot systems, is to develop an efficient and robust algorithm which allows the systems to be self-organized, self-configurable, self-adaptive, and self-repairable. However, it is well known that it is difficult to predict the emerging behaviors from local interactions of the individual agents, neither is it trivial to design rules for local interactions to generate a desired global behavior (Nolfi and Floreano, 2000).

The challenging issues in multi-robot systems also exist in the realization of basic behaviors, such as dynamic task allocation, robot coordination, and team reasoning, etc. (Yang and Gu, 2004). Furthermore, distributed multi-robot systems are usually facing the scalability issue. The computation complexity often grows exponentially with the number of robots (Klavins, 2003). As an emerging field, multi-robot systems aim at providing both principles for the construction of complex systems involving multiple robots and mechanisms for coordination of independent robot's behaviors (Stone and Veloso, 2000).

Recently, biologically-inspired systems have attracted extensive attention to tackle the scalability issue for multi-agent systems while maintaining system robustness and individual simplicity. Among those systems, swarm intelligence-based methods are one of the most popular paradigms. Swarm intelligence is an innovative computational and behavioral metaphor for solving problems in a distributed way inspired by the behaviors of social insects, such as swarming, flocking, herding, and shoaling phenomena in vertebrates. The social insect colonies are able to build sophisticated structures and regulate the activities of millions of individuals by endowing each individual with simple rules based on local perceptions. More and more researchers have applied these swarm intelligence-based approaches to solve real-world problems using multi-robot systems (Dorigo et al., 1996; Jatmiko et al., 2007; Meng and Gan, 2007; Meng and Gan, 2008; Pugh and Martinoli, 2007; Werfel, 2004). Since the

behaviors of agents in most of these systems are based on some predefined heuristics, it is difficult to ensure that the system can achieve globally-optimal behaviors with fixed parameters under dynamic environments.

On the other hand, biological organisms have evolved to perform and survive in a world characterized by rapid changes, high uncertainty, infinite richness, and limited availability of information (Pfeifer et al., 2007). Complex biological organisms are constructed through natural evolution from simple cells. Gene Regulatory Networks (GRNs) play a central role in understanding natural evolution and development (Alon, 2006). GRNs are models of genes, and the interactions of gene products. Each GRN is a collection of DNA segments in a cell which interact with each other indirectly through their RNA, protein product, and other chemicals in the cell, therefore governing the rates at which genes in the network are transcribed into mRNA.

It is very challenging to gain a thorough understanding of complex patterns of behavior from the interactions between genes in a regulatory network. A large number of different GRN models have been proposed, such as directed graphs, Bayesian networks, Boolean networks, neural networks, ordinary and partial differential equations, and rule-based formalisms (de Jong, 2002; Endy and Brent, 2001; Hasty et al., 2001; McAdams and Arkin, 1998; Smolen et al., 2000).

As far as we know, very little work has been conducted on multi-robot systems using GRNs or cellular mechanisms. Shen et al. (2004) proposed a digital hormone model (DHM) as a bio-inspired distributed control method for robot swarms and self-organization. Essentially, they applied Turing's reaction-diffusion model (Turing, 1952) to describe the interactions between the hormones. The DHM integrated dynamic network, topological stochastic action selection, and distributed control by hormone reaction-diffusion. More recently, Taylor (2004) proposed a gene regulatory network inspired real-time controller for a group of underwater robots. Then a genetic algorithm (GA) was applied to evolve the controller for a simple clustering task.

In this paper, we propose a distributed gene regulatory network-based algorithm for a multi-robot system in a construction task. Through this algorithm, robots in a multi-robot system can organize themselves autonomously into different predefined shapes, and reorganize themselves adaptively under dynamic environments. In evolving the robot construction behaviors, two objectives need to be optimized. One is the travel distance from the initial position to the destination point on the shape, which is represented by robot trajectories. The second objective is the time for the robots to converge to the final evenly distributed shape. In other words, we want to minimize the system convergence time as well as the travel distance to the final construction form. Since these two objectives are conflicting with each other, we use Non-dominated Sorting Genetic Algorithm-II (NSGA-II) (Deb et al., 2002), a popular multi-objective optimization (MOO) algorithm to obtain a number of Pareto-optimal solutions. Our simulation results show that the proposed cellular model is able to generate a fast convergence to the desired shape with a short travel distance.

The major advantages of our proposed method compared to the above two methods (Shen et al. 2004; Taylor, 2004) are: (1) The system's global information, such as the shape function, can be embedded into the GRN dynamics directly; (2) The dynamics of the GRN model can automatically drive the robots to their target positions while avoiding collision between the robots and obstacles inside the environment; (3) An MOO algorithm is applied so that we can achieve multiple Pareto-optimal solutions that trade off between a shorter travel distance of each robot to the shape and the overall convergence time of the system. From these solutions, we are able to pick out one that best suits our need. (4) A theoretical proof of the system's convergence is provided with the parameter constraints to ensure that the multi-robot system can converge to the predefined shape.

The remainder of the paper is organized as follows. Section II introduces the proposed cellular mechanisms for multi-robot controller using a gene regulatory network. A theoretical proof of the system convergence is provided in Section III. The evolutionary algorithm is presented in Section IV to achieve the Pareto-optimal solutions which can trade off the robots' travel distance and system convergence time. To evaluate the proposed method, several case studies of a multi-robot system for different construction tasks are presented in Section V. Conclusion and future work are discussed in Section VI.

2. Cellular Mechanisms for Multi-Robot Construction

2. 1 Biological Background

When a protein-coding gene is expressed, information stored in an organism's genome is transcribed and translated into proteins. Some of these proteins are transcription factors that can regulate the expression of their own or other genes. Thus, these proteins are under regulatory control, resulting in complex networks of interacting genes. These gene regulatory networks control a number of important cellular processes including responding to the environment, regulating the cell cycle and guiding the development of an organism.

Ordinary differential equations (ODEs) have been used to model the reaction kinetics of regulatory systems with a long history. Biological processes are highly complex, and usually simplification assumptions are needed for most mathematical models of GRNs. The first assumption is that the control of gene expression resides in the regulation of gene transcription. The second is that genes are expressed and proteins are produced at a continuous rate (Geard, 2004). The major advantage of ODE models lie in their detailed representation of regulatory interactions that can provide a more accurate representation of the physical system under investigation. Furthermore, a large number of dynamical system theories can provide tools for model analysis. In this section, we will introduce a single-cell GRN model, then a multi-cell GRN model.

2.1.1 A Single-Cell GRN Model

In a simplified, single-cell, non-spatial, biological model, a cell consists of one genome and several types of proteins. A genome may consist of several genes that interact with each other through their produced proteins. Each gene has a regulatory region and a structural region. The regulatory region specifies the proteins that inhibit or activate its expression, while the structural region describes the proteins that are produced when the gene is expressed. Here, when a gene is expressed, it means that its expression value is over a certain threshold. Fig. 1 provides an example GRN with 2 genes, where the product of gene 1 regulates the expression of gene 2, and the product of gene 2 regulates the expression of gene 1.

For a single cell, the expression of a gene with auto-regulation can be described by the following differential equations (Jin and Sendhoff, 2008a):

$$\frac{dg_i}{dt} = -\gamma_g g_i + \alpha_g f(p_i) \tag{1}$$

$$\frac{dp_i}{dt} = -\gamma_p p_i + \alpha_P g_i \tag{2}$$

where g_i is the expression level (measured by the concentration of its mRNA product) of gene i and p_i is the concentration of protein i. γ_g and γ_P are the decay rate of mRNA and protein concentration, respectively. α_g and α_p are the synthesis rate of mRNA and protein concentration, respectively. $f(x)$ is a sigmoid function, which can be defined as follows in case of auto-repression:

$$f(x) = \frac{\beta}{\theta^n + x^n} \tag{3}$$

where β is the activation coefficient, θ is the threshold, n is known as the Hill coefficient.

2.1.2 A Multi-cellular GRN Model

In a multi-cellular organism, it is necessary to model the intercellular communications. In addition to the internal dynamics of the cell, we should also consider external factors such as morphogen gradients, transcription factors diffused from other cells, and physical interactions between cells into the GRN model. Turing (1952) proposed one of the earliest models for pattern formation, where a pair of coupled reaction-diffusion equations was proposed to describe a system consisting of two morphogens. As two morphogens diffuse across a spatial field and react with each other, a variety of patterns emerge depending on parameter values. The gradients of protein concentrations across cells are a critical feature in embryonic development. The reaction-diffusion equations have been widely used in mathematical biology to study pattern formation in development (Gierer, 1981; Kauffman, 1993; Maini et al., 1997).

Salazar-Ciudad et al. (2000) proposed a GRN model with reaction-diffusion mechanism as follows:

$$\frac{dx_{ij}}{dt} = f_j(\mathbf{x}_i, \mathbf{u}) \quad \gamma_i x_{ij} \quad D_j \quad {}^2 x_{ij}, \quad 1 \quad i \quad n, 1 \quad j \quad m \tag{4}$$

where x_{ij} is the concentration of gene product j in cell i. The first term specifies the production of x_{ij}, the second term is its degradation, and the last term specifies the diffusion component at diffusion rate D_j. f_j is a nonlinear update function of gene product j, which is usually defined as a sigmoid function as $f(x) = \frac{1}{1+e^x}$. $\mathbf{u}$ is the vector of external input signals. γ_i is the degradation rate of product i. n is the number of gene products, and m is the number of cells.

2.2. A Distributed GRN-Based Algorithm for Multi-Robot Construction

The objective of multi-robot construction is to deploy multiple robots uniformly on a predefined two-dimensional (2D) shape, for example: a circle, a square, or any kind of arbitrary shapes, through a distributed control approach. Each robot only knows its local information without any global observer.

Analogous to biological systems, in our GRN-based control model, it is assumed that each robot corresponds to a single cell. Within each cell's genome, there are two genes, one for x-position and the other for y-position in a 2D environment. Each gene can produce a certain protein. Each protein can provide the following three functions: (1) To regulate the expression of the gene that produces it (i.e. auto-regulation); (2) To be able to diffuse proteins to its neighbors to avoid collision with other robots; (3) To adjust the robot's behaviors.

Inspired by Equations (1)-(4), the system dynamics of the GRN for multi-robot construction are defined as:

$$\begin{aligned} \frac{dg_{i,x}}{dt} &= -a \quad z_{i,x} + m \quad p_{i,x} \\ \frac{dg_{i,y}}{dt} &= -a \quad z_{i,y} + m \quad p_{i,y} \end{aligned} \tag{5}$$

$$\begin{aligned} \frac{dp_{i,x}}{dt} &= -c \quad p_{i,x} + k \quad f(z_{i,x}) + b \quad D_{i,x} \\ \frac{dp_{i,y}}{dt} &= -c \quad p_{i,y} + k \quad f(z_{i,y}) + b \quad D_{i,y} \end{aligned} \tag{6}$$

where $g_{i,x}$ and $g_{i,y}$ are the expression level of the genes for x-position and y-position of i^{th} robot, respectively. $p_{i,x}$ and $p_{i,y}$ are the concentration of the i^{th} robot's proteins produced by the x-position and y-position genes, respectively.

In order to embed the predefined 2D shape, which is the global information, into the regulatory dynamics, we define $f(z_i)$ as the following sigmoid functions:

$$f(z_{i,x}) = \frac{1-e^{-z_{i,x}}}{1+e^{-z_{i,x}}}$$
$$f(z_{i,y}) = \frac{1-e^{-z_{i,y}}}{1+e^{-z_{i,y}}} \quad (7)$$

where $z_{i,x}$ and $z_{i,y}$ are the gradients along x-axis and y-axis, respectively, of a predesigned function h at the robot's current gene expression level, which are defined as:

$$z_{i,x} = \frac{h}{g_{i,x}}, \quad z_{i,y} = \frac{h}{g_{i,y}} \quad (8)$$

where the predesigned function h is the function of the desired shape where robots are supposed to be deployed uniformly. We can also treat function h as the predefined gradient for cell migration. To facilitate the generation of the desired dynamics, we defined h as the square of the desired shape function. For example, if we want to deploy the robots onto a unit circle, the shape function can be defined as

$$s(g_{i,x}, g_{i,y}) = g_{i,x}^2 + g_{i,y}^2 - 1 = 0 . \quad (9)$$

Then function h can be defined as

$$h = (g_{i,x}^2 + g_{i,y}^2 - 1)^2 . \quad (10)$$

D_i is defined as the protein diffusion which aims at keeping the robot away from its neighbors. The size of neighborhood varies according to different shapes and different number of robots. In case of a circular shape, the neighborhood size can be defined as $\frac{2\pi r}{N}$, where r is the radius of the circle, and N is the total number of robots which are expected to deploy on the circle.

When a robot detects its neighbor, it will receive the protein emitted from that neighbor so that it can keep itself away from that neighbor to avoid collision. After summing up all the neighbors' diffused protein, we have

$$D_{i,x} = \sum_{j=1}^{N_i} D_{i,x}^j, \quad D_{i,y} = \sum_{j=1}^{N_i} D_{i,y}^j \quad (11)$$

where N_i denotes the number of its neighbors, and $D_{i,x}^j$ and $D_{i,y}^j$ are protein concentrations diffused from robot j into robot i, which is defined as

$$D_{i,x}^j = \frac{(g_{i,x} - g_{j,x})}{\sqrt{(g_{i,x} - g_{j,x})^2 + (g_{i,y} - g_{j,y})^2}} \quad (12)$$

$$D_{i,y}^j = \frac{(g_{i,y} - g_{j,y})}{\sqrt{(g_{i,x} - g_{j,x})^2 + (g_{i,y} - g_{j,y})^2}} \quad (13)$$

where the directions of $D_{i,x}^j$ and $D_{i,y}^j$ are defined to be from robot j to robot i along x-axis and y-axis, respectively.

Initially, the robots are located randomly in a 2D space. By following the dynamics defined in Equations (5) and (6), eventually multiple robots can be deployed uniformly on the predefined shape automatically. In other words, the system can be stabilized to an equilibrium state defined by the shape. Essentially, the shape information is the global information, which can be elegantly embedded into the dynamics of each individual robot through function $f(\mathbf{z}_i)$.

From Equations (5) and (6), we can see that other robots' impact on a particular robot is exercised through the diffusion of proteins, which is implemented by D_i. Each robot itself has two state vectors: gene expression g_i and protein concentration p_i. These two variables regulate each other via positive or negative feedback. Through these coherent links, we can achieve the goal of deploying robots to form a certain shape. In general, each robot always has to balance two forces: one force is to approach the predefined shape as provided inside the gene expression, and the other one is to avoid collision with each other as implemented via protein diffusion.

3. Theoretical Analysis of System's Convergence

In this section, a theoretical proof of the system's convergence to the predefined shape using the proposed GRN-based algorithm is provided. Considering Equations (5) and (6), the protein diffusion among the neighboring robots may counteract with each other. For simplicity, we can neglect the dynamics of the protein diffusion in the proof of the system convergence. Therefore, Equations (5) and (6) can be rewritten as follows:

$$\frac{dg_x}{dt} = -a\ z_x + m\ p_x$$
$$\frac{dg_y}{dt} = -a\ z_y + m\ p_y \quad (14)$$

$$\frac{dp_x}{dt} = -c\ p_x + k\ f(z_x)$$
$$\frac{dp_y}{dt} = -c\ p_y + k\ f(z_y) \quad (15)$$

Here, we remove the subscript i since every robot shares the same dynamics.

Before proving the system convergence, we would like to propose and prove the following Lemma first.

Lemma 1: For the activation function defined in Equation (7), $|f(x)| \ |x|$ holds for all x.

Proof: Since $f(x)$ is a sigmoid function with the analytical form $f(x) = \frac{1-e^{-x}}{1+e^{-x}}$, we can draw the following conclusions: (1) $|f(x)| \ 1$; and (2) $0 < f'(x) < 1$. Refer to the Appendix for the detailed proofs of the these two conclusions.

When $x = 0$, $f(x) = x = 0$. When $x \ 0$, according to the integration median theory, we can get $f(x) = f'(\varepsilon) \ x$. Therefore, from $0 < f'(x) < 1$, we can have $|f'(\varepsilon)| < 1$, so $|f(x)| = |f'(\varepsilon) \ x| < |x|$. Thus, we have $|f(x)| \ |x|$ and $|f(x)| = |x|$ only if $x = 0$.

Theorem 1: The state vectors g_x and g_y in Equation (5) will converge to the target shape defined by Equation (9), and the state vectors p_x and p_y in Equation (6) will converge to zero provided that $m\ k\ \ a\ c$ and $k, c, a, m > 0$.

Proof:

According to Lyapunov Theory, we can claim that the system defined by Equations (14) and (15) will be convergent if we can find a Lyapunov function $V(g_x, g_y, p_x, p_y)$ that satisfies the following conditions:

1. $V(g_x, g_y, p_x, p_y)$ *is positive definite;*
2. $V(g_x, g_y, p_x, p_y)$ *is negative definite.*

There are four parameters, *a, m, c,* and *k* in the GRN-based dynamics model. First, all parameters must be positive. Then we construct a parameterized energy function of the system dynamics in the following form:

$$V(g_x, g_y, p_x, p_y, s) = h(g_x, g_y) + \frac{1}{2} s\ \ p_x^{\ 2} + \frac{1}{2} s\ \ p_y^{\ 2} \quad (16)$$

Now we follow the steps of Lyapunov theory to prove the system's convergence.

(1) $V(s)\ \ 0$.

(2) $\frac{dV}{dt} = \frac{dh}{dt} + s\ \ p_x\ \ \frac{dp_x}{dt} + s\ \ p_y\ \ \frac{dp_y}{dt}$

$= \frac{h}{g_x}\ \ \frac{dg_x}{dt} + \frac{h}{g_y}\ \ \frac{dg_y}{dt} + s\ \ p_x\ \ \frac{dp_x}{dt} + s\ \ p_y\ \ \frac{dp_y}{dt}$ (consider Equations (14) and (15))

$= \frac{h}{g_x}\ (-a\ \ \frac{h}{g_x} + m\ \ p_x) + \frac{h}{g_y}\ (-a\ \ \frac{h}{g_y} + m\ \ p_y)$

$+ s\ \ p_x\ (-c\ \ p_x + k\ \ f(\frac{h}{g_x})) + s\ \ p_y\ (-c\ \ p_y + k\ \ f(\frac{h}{g_y}))$

(since $z_x = \frac{h}{g_x}$ and $z_y = \frac{h}{g_y}$)

$= -a z_x^{\ 2} - a z_y^{\ 2} - c\ \ s\ \ p_x^{\ 2} - c\ \ s\ \ p_y^{\ 2}$

$+ m\ \ p_x\ \ z_x + m\ \ p_y\ \ z_y + k\ \ s\ \ p_x\ \ f(z_x) + k\ \ s\ \ p_y\ \ f(z_y)$

$- a\ \ z_x^{\ 2}\ \ a\ \ z_y^{\ 2}\ \ c\ \ s\ \ p_x^{\ 2}\ \ c\ \ s\ \ p_y^{\ 2}\ \ m\ |p_x|\ \ |z_x|$

$+ m\ |p_y|\ \ |z_y| + k\ \ s\ |p_x|\ \ |f(z_x)| + k\ \ s\ |p_y|\ \ |f(z_y)|$

(from Lemma 1, we can get $|f(z)|\ \ |z|$)

$- (\sqrt{a} z_x\ \ \sqrt{c\ \ s}\ \ p_x)^2\ \ (\sqrt{a} z_y\ \ \sqrt{c\ \ s}\ \ p_y)^2$

$-2\sqrt{a\ \ c\ \ s}\ |z_x|\ \ |p_x| - 2\sqrt{a\ \ c\ \ s}\ |z_y|\ \ |p_y|$

$+ (m + k\ \ s)\ |p_x|\ \ |z_x| + (m + k\ \ s)\ |p_y|\ \ |z_y|$

$= -(\sqrt{a}\ \ z_x - \sqrt{c\ \ s}\ \ p_x)^2 - (\sqrt{a}\ \ z_y - \sqrt{c\ \ s}\ \ p_y)^2$

$+ (k\ \ s - 2\sqrt{a\ \ c\ \ s} + m)\ (|p_x|\ \ |z_x| + |p_x|\ \ |z_x|)$

$= -(\sqrt{a}\ \ z_x - \sqrt{c\ \ s}\ \ p_x)^2 - (\sqrt{a}\ \ z_y - \sqrt{c\ \ s}\ \ p_y)^2$

$+ k\ \ [(\sqrt{s} - \frac{\sqrt{a\ \ c}}{k})^2 - \frac{a\ \ c}{k^2} + \frac{m}{k}]\ \ (|p_x|\ \ |z_x| + |p_y|\ \ |z_y|)$

Let $s = \frac{a\ \ c}{k^2}$, we can get

$\frac{dV}{dt} - (\sqrt{a}\ \ z_x\ \ \sqrt{c\ \ s}\ \ p_x)^2\ \ (\sqrt{a}\ \ z_y\ \ \sqrt{c\ \ s}\ \ p_y)^2$

$+ k\ \ (-\frac{a\ \ c}{k^2} + \frac{m}{k})\ \ (|p_x|\ \ |z_x| + |p_y|\ \ |z_y|)$

So if $-\frac{a\ \ c}{k^2} + \frac{m}{k}\ \ 0$,

which can be rewritten as $m\ \ k\ \ \ a\ \ c$ we can ensure $\frac{dV}{dt}\ \ 0$.

Since both conditions of the Lyapunov function *V* have been satisfied, we can claim that the system is convergent. Now we need to analyze whether the system will converge to the target shape. As we know that the system will be stabilized at the points that satisfy $\frac{dV}{dt} = 0$.

If $\frac{dV}{dt} = 0$, all the 'less than ( )' conditions should satisfy the 'equal (=)' condition, which means that $|f(z)|\ \ |z|$ must be $|f(z)| = |z|$. From Lemma 1, the condition $|f(z)| = |z|$ holds only when $z = 0$. Therefore, if $\frac{dV}{dt} = 0$, we have $z_x = 0$ and $z_y = 0$ which means that $\frac{h}{g_x} = \frac{h}{g_y} = 0$. Since we define *h* function as $h(g_x, g_y) = s(g_x, g_y)^2$, here $s(g_x, g_y)$ is the shape function. $\frac{h}{g_x} = \frac{h}{g_y} = 0$ means $s(g_x, g_y) = 0$. Therefore, we can say that when $\frac{dV}{dt} = 0$, the system will converge to the target shape.

After verifying the system convergence, we further consider the uniform distribution problem. From Lyapunov theory, we can only ensure that robots can finally converge to the target shape. The uniform distribution is implemented by pre-specifying the protein diffusion range. Generally, we define the protein diffusion range to be $\frac{L}{N}$, where *L* is the length of the desired curve and *N* is the number of robots in the system.

4. The Evolutionary Algorithm

In addition to achieving convergence, the system performance can be further evaluated with the following two objectives: the total traveling distance of all the agents and the system convergence time. This is an MOO problem, where the objective function is no longer a scalar

value, but a vector. As a consequence, a number of Pareto-optimal solutions should be achieved instead of one single solution. Pareto-optimality is an important concept in multi-objective optimization and learning (Deb, 2001; Jin and Sendhoff, 2008b). Let us formally introduce the following basic concepts. Consider the following *m*-objective minimization problem (Jin and Sendhoff, 2008b):

$$\min F(X)$$

$$F = \{f_1(X), f_2(X), ..., f_m(X),\}.$$

Definition 1: A solution X is said to *dominate* a solution Y if $\forall j = 1, 2, ..., m,\ f_j(X) \quad f_j(Y)$, and there exists $k \quad \{1, 2, ...m\}$ such that $f_k(X) < f_k(Y)$.

Definition 2: Solution X is called *Pareto-optimal* if it is not dominated by any other feasible solutions.

Definition 3: Pareto front: There often exists more than one Pareto-optimal solution if the objectives are conflicting with each other. The image in the objective space composed of the Pareto-optimal solutions is known as the *Pareto front.*

In this paper, NSGA-II (Deb et al., 2002) has been adopted for evolution, which is a popular and efficient evolutionary algorithm for solving multi-objective optimization problems. In our work, simulated binary crossover (SBX) (Deb and Agrawal, 1995) and polynomial mutation (Deb and Goyal, 1996) have been employed to generate offspring. After the offspring population is generated, the elitist crowded non-dominated sorting is used for selecting parents for the next generation.

Different from single objective optimization algorithms, where often only one optimal solution is achieved, NSGA-II produces a set of Pareto-optimal solutions, i.e. in our case, the solutions that balance the convergence time and travel distance of the robots to the target shape. We will analyze the solutions when discussing the simulation results using NSGA-II.

5. Experiments and Results

To evaluate the reliability and the efficiency of the proposed method, several case studies for multi-robot construction are performed. We implement all the case studies using MATLAB. It is assumed that a 4 x 4 grid map is used in our simulation. Therefore, the unit of travel distance is the meter and the unit of convergence time is the second.

5.1 Multi-Objective Optimization of the GRN

Experimental setup. Five parameters in Equation (5) and (6), i.e., *a, m, c, k*, and *b*, need to be optimized using the NSGA-II. The goal of the optimization is to minimize the robots' travel distance and the convergence time, while assuring the system's stability.

For the simulation, we set the number of robots to be 8, and the population size for NSGA-II to be 100. The parameters of the NSGA-II were set as follows, as recommended by Deb et al (1995). The crossover probability was set to 0.9 and the distribution index for the SBX crossover was 20. Mutation probability was set to be inversely proportional to the number of the decision variables, which was 5 in our case, therefore, the probability was set to be 0.2 and the distribution index for mutation was set to be 20. The simulation was run for 50 generations, which is small but sufficient for this relatively small optimization problem. Initially, five parameters *k, c, b, a,* and *m* were assigned to random numbers ranging from 1 to 100.

Here, we defined the shape as a unit circle centered at (0, 0). The robot system consisted of 8 robots randomly distributed in the environment. During the deployment, the robots should approach to the unit circle and meanwhile avoid colliding with each other.

Since our final goal of the multi-robot system is to evenly distribute the robots on a predefined shape, the distance between the final positions of the robots and the target shape, termed position error hereafter, should be as small as possible. Therefore, we defined a threshold for the average position error as a constraint of the bi-objective problem when using the NSGA-II method.

In the following experiments, we set the threshold for the average position error for the circular shape to be $\frac{1}{r\ N}$, where *r* is the radius of the circle, and *N* is the total number of robots to be deployed on the circle.

Experimental results. We have performed 35 independent runs for optimizing the parameters using the NSGA-II. We found that the differences in the Pareto sets from the runs are minor and therefore, we presented here the Pareto optimal solutions from one typical run. In the optimization, both the total travel distance and the system's convergence time are minimized, subject to the constraint of the average position error. The non-dominated solutions from the last generation (50^{th}) of the evolutionary run are shown in Fig. 2. The robot trajectories of three solutions picked out from the Pareto-front solutions in Fig. 2 are shown in Fig. 3(a) through (c), respectively. The corresponding values of the parameter setup for these three solutions are listed in Table 1.

From Fig. 3, it can be seen that there is no significant difference between three solutions from the travel distance point of view. The major difference lies in the convergence time, where the solution (c) has the shortest convergence time. From Fig. 2, we can see that if we want to minimize the total traveling distance, we have to sacrifice the system's convergence time, and vice versa. This makes good sense from the control theory point of view that there is a trade-off between the system's response time and convergence time. Fig. 2 will be helpful for users to design the system parameters based on their specific requirements. For example, if the user concerns more about the convergence time, she or he may prefer Pareto-optimal solutions close to (c). If the user has more concerns on total traveling distance to save energy, she or he may choose the Pareto-optimal solutions close to (a).

Analysis. To understand the inherent correlation between the system's parameters, i.e., k, c, b, a, m, and two performance objectives, i.e. total travel distance and

system's convergence time, *canonical correlation analysis* method (Mardia et al., 1979) is adopted here. Canonical correlation analysis (Becker et al. 1988) aims to find a certain linear combination that can maximize the correlation of a vector X and another vector $\boldsymbol{Y}$. In our case, vector $\boldsymbol{X}$ can be defined as a linear combination of parameters, k, c, b, a, m, and vector $\boldsymbol{Y}$ can be defined as a linear combination of the performance indices, *i.e., travel distance* and *convergence time*.

After calculation, we get that the coefficient of input space X is *xcoef* = [-0.0017045001, 0.0005503515, 0.0008908889, -0.0169773006, 0.0081786459] and the coefficient of output space Y is $ycoef = [-0.77476111, -0.07554193]$. It means that X is a linear combination of $xcoef$ best influences Y a linear combination of $ycoef$. The correlation between $xcoef$ X and $ycoef$ Y is 0.9990042. Fig. 4 shows the relationship between input $xcoef$ X and output $ycoef$ Y. From Fig.4, we can see that $xcoef$ X (i.e., the linear combination of k, c, b, a, m) and $ycoef$ Y (i.e., the linear combination of *dist* and *time*) is nearly linearly correlated.

Another way to analyze the relationship between the parameters of the system and the system's performance is to use a combination of parameters of the dynamical system based on the convergence proof in Equations (14) and (15). Since $\frac{dV}{dt} = -(\frac{ac-km}{k}) \mid p \mid \mid \frac{f}{g} \mid$, if we increase the value of *(ac-km)/k* , the absolute value of $\frac{dV}{dt}$ will increase accordingly. In this case, the robots will travel faster, but the travel distance will become longer, though the convergence time will be shorter. By contrast, a decrease in the value of *(ac-km)/k* will lead to a shorter travel distance and a longer convergence time. Figs. 5 shows all the Pareto-optimal solutions with respect to *(ac-km)/k*, and the relationship between the total travel distance and *(ac-km)/k,* and the relationship between the convergence time and *(ac-km)/k* are shown in Fig. 6 and Fig. 7, respectively. These results confirm the correlation relationship revealed from the theoretical analysis.

The evolutionary optimization has been performed when the number of robots is set to 8 and the predefined shape is a circle. In the following, we are going to present the results of a few case studies to check the system's performance when the number of robots is changed or when the desired shape is changed. In addition, we also demonstrate system's self-organization and self-adaptation ability in case new robots join after deployment, and when obstacles are present on the desired shape, or even when the obstacles are moving in the environment. Without loss of generality, we pick out solution (c), which is optimized for minimizing the convergence time, from the Pareto-optimal solutions (refer to Fig. 2) for the following experiments.

5.2 Case Study 1: Change in the Number of Robots

Here, we aim to deploy a number of initially randomly distributed robots to a unit circle centered at (0, 0).. During the deployment, the robots should approach the unit circle while avoiding colliding with one another. Three setups have been used to evaluate the proposed algorithms, where the number of robots is 5, 10 and 20, respectively. The trajectories of the robots of the three setups are shown in Fig. 8(a), 8(b), and 8(c), respectively. We performed 35 independent runs for each setup and the results of the system's performance for each setup are listed in Table 2.

From Fig. 8 and Table 2, we can see that for all these three setups, randomly initiated robots can ultimately deployed on the desired shape uniformly with our proposed method. Although we use the Pareto-optimal solution obtained from the 8-robot setup, the simulation results demonstrate that system's performance in terms of both the total travel distance and convergence time is satisfactory when the number of robots is changed. It means that the Pareto-optimal solution obtained from NSGA-II method is scalable to the number of robots. Furthermore, from Table 2, it can be seen that the position errors are very small in all setups.

5.3 Case Study 2: Change in the Target Shape

In this case, we plan to deploy a number of initially randomly distributed robots to a unit square. The unit square is defined as follows: lower-left point at (-0.5,-0.5) and the upper-right point at (0.5, 0.5). It is a little bit tricky to define the shape function s to be a unit square instead of a unit circle. We first set up a circle of a radius of $\sqrt{1/2}$ centered at (0, 0), which can be defined as:

$$s_1(g_{i,x}, g_{i,y}) = g_{i,x}^2 + g_{i,y}^2 - \frac{1}{2} = 0$$

and $f_1 = (s_1(g_{i,x}, g_{i,y}))^2$. With help of this function, we can deploy the robots on the specified circle. Then we define $s_2(x, y)$ as follows:

$$s_2(g_{i,x}, g_{i,y}) = \begin{cases} (g_{i,x} + 1/2) \\ (g_{i,x} - 1/2) \\ (g_{i,y} + 1/2) \\ (g_{i,y} - 1/2) \end{cases}$$

$$\text{if} \quad \begin{array}{l} g_{i,x} - 0 \quad \text{and} \quad 1/2 \quad g_{i,y} \quad 1/2 \\ g_{i,x} - 0 \quad \text{and} \quad 1/2 \quad g_{i,y} \quad 1/2 \\ -1/2 \quad g_{i,x} \quad 1/2 \quad \text{and} \quad g_{i,y} \quad 0 \\ -1/2 \quad g_{i,x} \quad 1/2 \quad \text{and} \quad g_{i,y} \quad 0 \end{array} \tag{17}$$

$$f_2 = (s_2(g_{i,x}, g_{i,y}))^2 \tag{18}$$

With the help of this function, the robots will be able to converge to the defined unit square.

We conducted two experiments to deploy the robots onto the unit square, where the number of robots is 8 and 12, respectively. Fig. 10 shows the simulation results of deploying 8 robots to a unit square.. The simulation results for 12 robots are similar. The convergence time for 8-robot case is 0.84 and the total travel distance is 29.03. It can be seen from Fig. 10 that robots can automatically deploy

themselves evenly on the square. This case study indicates that the proposed GRN-based control method can be used to control multiple robots to deploy on different shapes as long as the shape function can be defined analytically.

5.4. Case Study 3: Self-Reorganization

From the previous two case studies, we can see that the proposed GRN-based control algorithm can automatically drive multiple robots to a predefined shape. In this case study, we intend to evaluate the system's capability of self-reorganization. More specifically, we want to investigate whether the robots can self-organize themselves when new robots join the team.

Fig. 11 provides the trajectories of the robots during the initial shape formation as well as those during the self-reorganization to incorporate the newcomers. Fig. 11 demonstrates that the robots are able to self-reorganize themselves when newcomers join the team even after the deployment is already complete.

5.5. Case Study 4: Self-Adaptation to a Static Obstacle

Another interesting test for our algorithm is whether our algorithm can self-adapt to the environment containing static obstacles on the predefined shape., For this purpose, an obstacle is set on the perimeter of the desired shape. We then study if the robots can circumvent the obstacle and continue to form the desired shape using the proposed method.

Fig. 12 shows the simulation results of three different multi-robot systems, i.e., where the number of robots is set to 5, 8, and 10, respectively, and one obstacle is located on the perimeter of the target shape. Again, 35 independent runs are conducted for the three cases, and the results are listed in Table 3. Here, the obstacle is modeled as a static robot. The obstacle can emit protein to neighboring robots so that it can influence neighboring robots' behaviors. However, it does not receive protein diffusion and it cannot move.

From Fig. 12, we can see that the multi-robot systems can adapt themselves to the environment change. They are supposed to form a circle, since there is an obstacle, they can circumvent the obstacle and form a circle on the rest of the perimeter. From Table 3, it can be seen that the position errors of the robots are very small in all cases.

5.6. Case Study 5: Self-Adaptation to Moving Obstacles

In this case study, we aim to test whether the proposed algorithm for multi-robot systems can be self-adaptive to moving obstacles in the environment, i.e., whether the robots can move to the target shape while avoiding collisions with moving obstacles in the environment.

Fig. 13 is a set of snapshots of robots' adaptive behaviors, where the robots aim to move to a unit circle while avoiding two moving obstacles. In the experiment, obstacle 1 moves from bottom-left to top-right and obstacle 2 moves from bottom-right to top-left. In Fig. 13(a), the robots are already distributed on a circle and two obstacles are approaching the robots. In Fig. 13(b), obstacle 2 moves faster than obstacle 2, and obstacle 2 is approaching toward the robots. We can see that some robots start to avoid obstacle 2. In Fig. 13(c), robots try to avoid both obstacles. In Fig. 13(d), obstacle 2 moves away from the robots, while the robots try to avoid obstacle 1. In Fig. 13(e), the robots try to avoid obstacle 1, when obstacle 2 has left the simulation environment. In Fig. 13(f), obstacle 1 moves away from the robots. Although only two obstacles are implemented in our experiment here, similar behaviors are expected for multiple moving obstacles as well.

6. Conclusion and Future Work

In this paper, we have presented a novel GRN-inspired distributed control approach to multi-robot construction. Compared to other multi-robot control methods, the major merits of the proposed method are: (1) embedding the global shape information into the regulatory dynamics through a sigmoid function; (2) truly distributed behaviors of each robot balanced by two different forces: one force to approach to the predefined shape and the other to avoid collision with other robots. The local interaction among the robots is represented by the diffusion terms in the regulation dynamics inspired from multi-cellular mechanisms of living organisms.

In the future, we will continue our research on GRN-inspired multi-robot controllers. We will investigate arbitrary shape construction by evolving a GRN model. We will also investigate the system's robustness when one or more robots fail.

It is always an interesting question to ask whether we can gain additional insights into biological systems from what we have learned in a successful biologically-inspired engineering system. In our multi-robot system, we have shown that the global shape information of the robots can be gracefully embedded into a gene regulatory network model, and the regulatory dynamics can ensure the convergence of the robots into the desired shape in the presence of uncertainties. This finding may help us in computational modeling of pattern formation (Schnabel et al, 2006) in the early development of biological organisms, in particular regarding the role of planar cell polarity signaling and sensory feedback of ciliated cells (Simon and Mlodzik, 2008; Singla and Reiter, 2006).

Appendix

Assume that $f(x)=\frac{1-e^{-x}}{1+e^{-x}}$, which is a sigmoid function.

Conclusion 1: $|f(x)|$ 1

Proof:

$$f(x)=\frac{1-e^{-x}}{1+e^{-x}}=-1+\frac{2}{1+e^{-x}},$$

$$e^{-x}>0 \quad 1+e^{-x}>1 \quad 0<\frac{2}{1+e^{-x}}<2$$

Since

$$- \quad <1- \quad +1 \quad \frac{2}{1+e^{-x}} \quad 1$$

We have $-1<f(x)<1$, which means $|f(x)|$ 1.

Conclusion 2: $0 < f^{'}(x) < 1$.

Proof:

Apply derivative on $f(x)$, we have

$$f^{'}(x) = 2 * \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1-f^2(x)}{2}$$

Since $-1 < f(x) < 1$, we can get $0 < f^{'}(x) < \frac{1}{2} < 1$.

References

Alon, U., 2006. An Introduction to Systems Biology: Design Principles of Biological Circuits. Chapman & Hall/CRC Mathematical and Computational Biology.

Becker, R. A., Chambers, J.M. and Wilks, A. R. 1988. The New s Language: A Programming Environment for Data Analysis and Graphics. Wadsworth and Brooks/Cole Computer Science Series.

Deb, K., 2001. Multi-Objective Optimization using Evolutionary Algorithms. John Willy & Sons, Ltd

Deb, K. and Agrawal, R.B, 1995. Simulated Binary Crossover for Continuous Search Space. Complex Systems, Vol. 2, No. 9:115-148.

Deb, K. and Goyal, M., 1996. A Combined Genetic Adaptive Search(GeneAS) for Engineering Design. Computer Science and Informatics, Vol 4, No.26: 30-45

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2: 182-197.

De jong, 2002. H. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. Journal of Computational Biology, vol. 9, No. 1: 67-103.

Dorigo, M., Maniezo, V., Colorni, A., 1996. The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man and Cybernetics-Part B, Vol. 26, No. 1: 29-41.

Endy, D. and Brent, R., 2001. Modeling cellular behavior. Nature 409, 391-395.

Geard, N., 2004 Modeling Gene Regulatory Networks: Systems Biology to Complex Systems, ACCS Technical Report.

Gierer, A., 1981. Generation of Biological Patterns and Form: Some Physical, Mathematical, and Logical Aspects. In Progress in Biophysics and Molecular Biology. Vol. 37, No. 1: 1-47.

Hasty, J., McMillen, D., Isaacs, F. and Collins J.J., 2001. Computational Studies of Gene Regulatory Networks: In Numero Molecular Biology. Nature Review Genetics 2, 268-279.

Jatmiko, W., Sekiyama, K. and Fukuda, T., 2007. A PSO-based Mobile Robot for Odor Source Localization in Dynamic Advection-Diffusion with Obstacles Environment: Theory, Simulation and Measurement, IEEE Computational Intelligence Magazine, Vol 2, Issue 2: 37-51.

Jin, Y. and Sendhoff, B., 2008a. Evolving *in silico* Bistable and Oscillatory Dynamics for Gene Regulatory Network Motifs, IEEE/CEC Congress on Evolutionary Computation, pp. 386-391, Hongkong, China.

Jin, Y. and Sendhoff, B. 2008b. Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies. IEEE Transactions on Systems, Man, and Cybernetics-part C: Applications and Reviews, Vol. 38, No.3: 397-415.

Kauffman, S. A., 1993. The Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press, New York.

Klavins, E., 2003. Communication Complexity of Multi-Robot Systems Springer Tracts in Advanced Robotics, Volume 7/2003 page 275-292.

Mardia, K. V., Kent, J. T. and Bibby, J.M., 1979 Multivariate Analysis (Probability and Mathematical Statistics). Academic Press, ISBN 0-12-471250-9.

Maini, P.K., Painter, K.J. and Nguyen, P.C., 1997. Spatial Pattern Formation in Chemical and Biological Systems. Journal of the Chemical Society, Faraday Transctions. Vol 93 No. 20: 3601-3610

McAdams, H. H. and Arkin, A., 1998. Simulation of Prokaryotic Genetic Circuits. Annual Review of Biophysics and Biomolecular Structure. Vol 27, 199-224.

Meng, Y. and Gan, J. 2007. LIVS: Local Interaction via Virtual Stigmergy Coordination in Distributed Search and Collective Cleanup. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, CA, USA.

Meng, Y. and Gan, J. 2008. A Distributed Swarm Intelligence based Algorithm for a Cooperative Multi-Robot Construction Task, IEEE Swarm Intelligence Symposium. St. Louis, Missouri.

Mjolsness, E., Sharp, D. H. and Reinitz, J., 1991. A Connectionist Model of Development. Journal of Theoretical Biology. Vol 152:429-453.

Nolfi, S. and Floreano, D. 2000. Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press, Cambridge, MA.

Pfeifer, R., Lungarella, M. and Iida, F., 2007. Self-Organization, Embodiment, and Biologically Inspired Robotics, Science, Vol. 318, No. 5853: 1088-1093.

Pugh, J. and Martinoli, A., 2007. Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. In: Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007).

Salazar-Ciudad, I., Garcia-Fernandez, H. and Sole, R.V., 2000 Gene Networks Capable of Pattern Formation: From Induction to Reaction-Diffusion. Journal of Theoretical Biology, Vol 205:587-603.

Schnabel, R., Bischoff, M., Hintze, A., Schulz, A.-K., Hejnol, A., Meinhardt, H., and Hutter, H., 2006. Global Cell Sorting in the *C. Elegans* Embrryo Defines a New Mechanism for Pattern Formation. Developmental Biology, Vol.294:418-431.

Shen, W., Will, P. and Galstyan, A., 2004. Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms, Autonomous Robots, Vol.17, Issue 1: 93-105.

Simons, M. and Mlodzik, M., 2008. Planar Cell Polarity Signaling: From Fly Development to Human Disease. Annu. Rev. Genet., Vol. 42:517-540.

Singla, V. and Reiter, J.F., 2006. The Primary Cilium as the Cell's Antenna: Signaling at a Sensory Organelle. Science, Vol. 313:629-633.

Smolen, P., Baxter, D.A. and Byrne, J.H., 2000. Modeling Transcriptional Control in Gene Networks: Methods, Recent Results, and Future Directions. Bull. Math. Biol. Vol 62, No. 2: 247-292.

Stone, P., Veloso, M., 2000 Multiagent Systems: A Survey from a Machine Learning Perspective. Autonomous Robots, Vol 8, No. 3:345–383.

Taylor, T., 2004. A Genetic Regulatory Network-Inspired Real-Time controller for a Group of Underwater Robots. In: Proceedings of Eighth Conference on Intelligent Autonomous Systems (IAS-8).

Turing, A. M., 1952. The Chemical Basis of Morphogenesis, Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, Vol 237, No. 641: 37-72

Werfel, J., 2004. Building Blocks for Multi-Robot Construction. In 7th International Symposium on Distributed Autonomous Robotic Systems (DARS 2004). Toulouse, France.

Yang, E. and Gu, D., 2004. Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey. Technical Report CSM-404, Department of Computer Science, University of Essex.

D_i

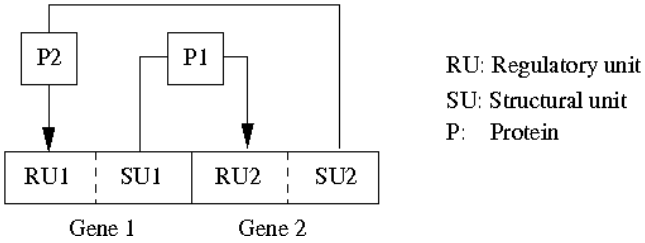


Fig. 1. Illustration of a GRN of a single cell with 2 genes.

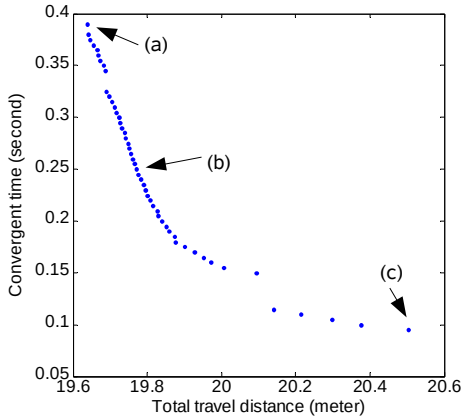
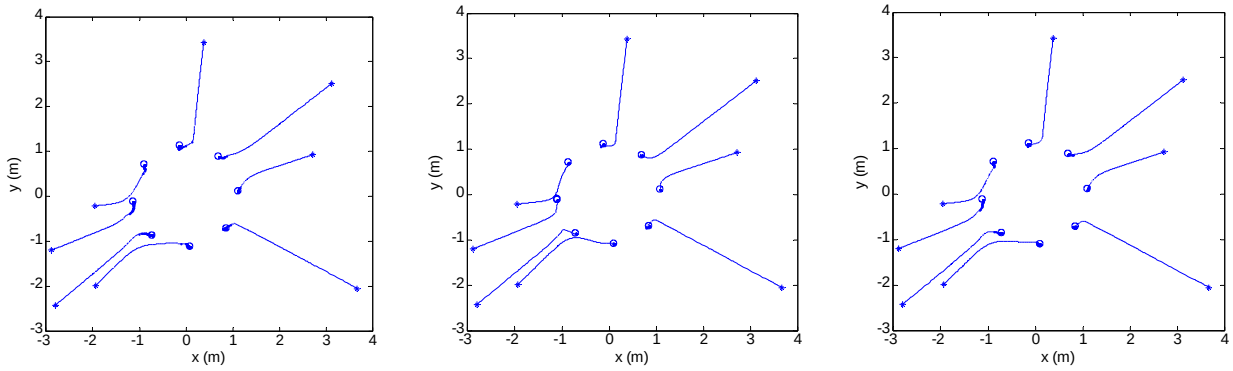


Fig. 2. The Pareto-optimal solutions using NSGA-II. The three solutions depicted in Fig. 3(a) through (c) are marked by the respective letters.



(a) (b) (c)

Fig. 3. The trajectories of 8 robots from initial positions (denoted by "*") to the final positions (denoted by "o") with different parameter solutions picked from the Pareto-front solutions marked in Fig. 2. a) Solution (a) has a long convergence time but a short travel distance, b) Solution (b) exhibits a medium convergence time, a relatively longer travel distance, c) Solution has a longer total travel distance but a much shorter convergence time.

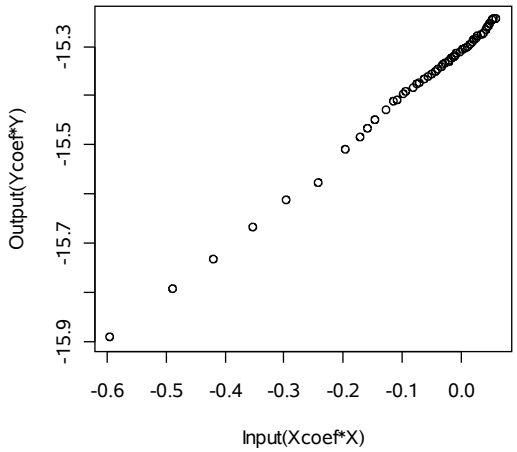


Fig. 4. The relationship between input parameter space $xcoef\ \ X$ and output performance space $ycoef\ \ Y$, where each circle represents one correlation solution.

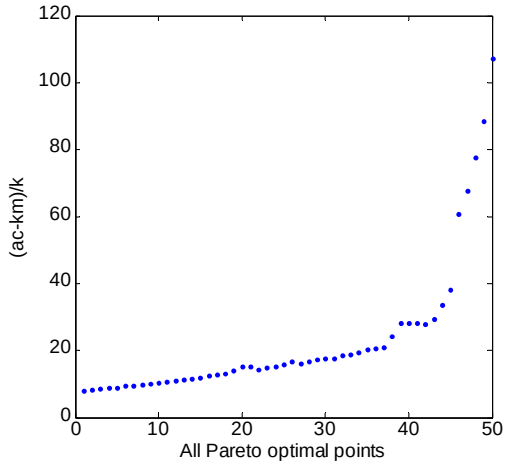


Fig.5. All the Pareto-optimal solutions plotted with respect to the parameter combination $(ac-km)/k$.

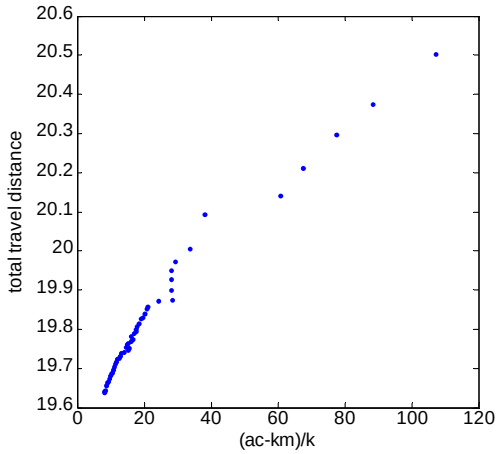


Fig.6. The total travel distance versus the parameter combination of $(ac-km)/k$ derived from theoretical analysis.

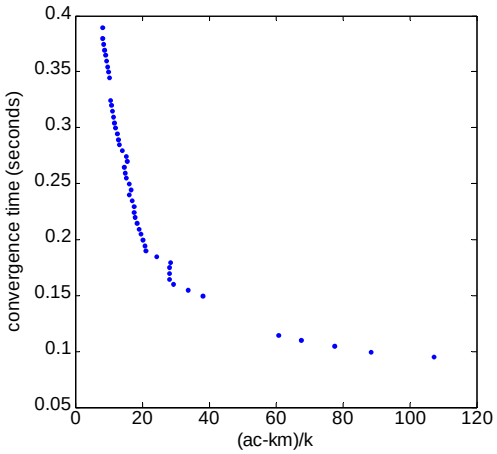
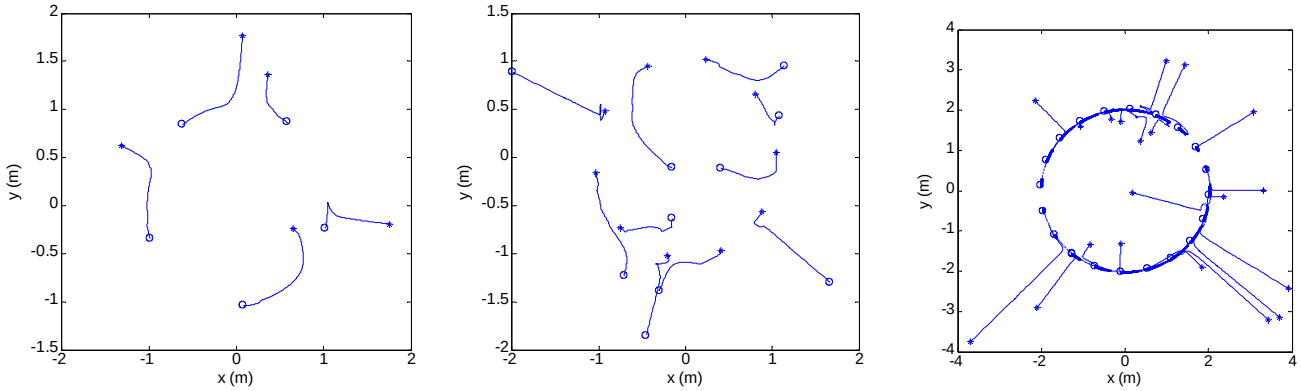


Fig.7. The convergence time versus the parameter combination of $(ac-km)/k$ derived from the theoretical analysis.



(a) (b) (c)

Fig. 8. Different groups of robots deployed uniformly on circles. Note that a unit circle is used for (a) and (b), and a circle with radius 2 is used for (c).

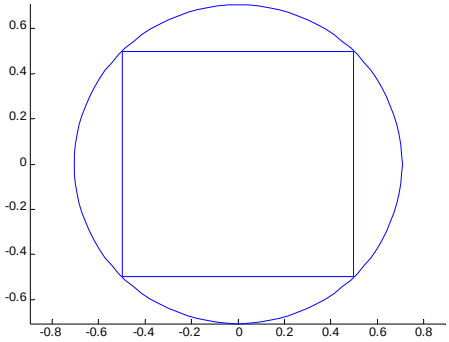


Fig. 9. The relationship between the circle and the desired square

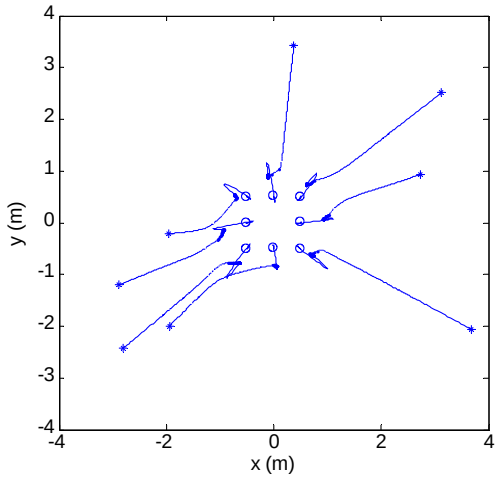
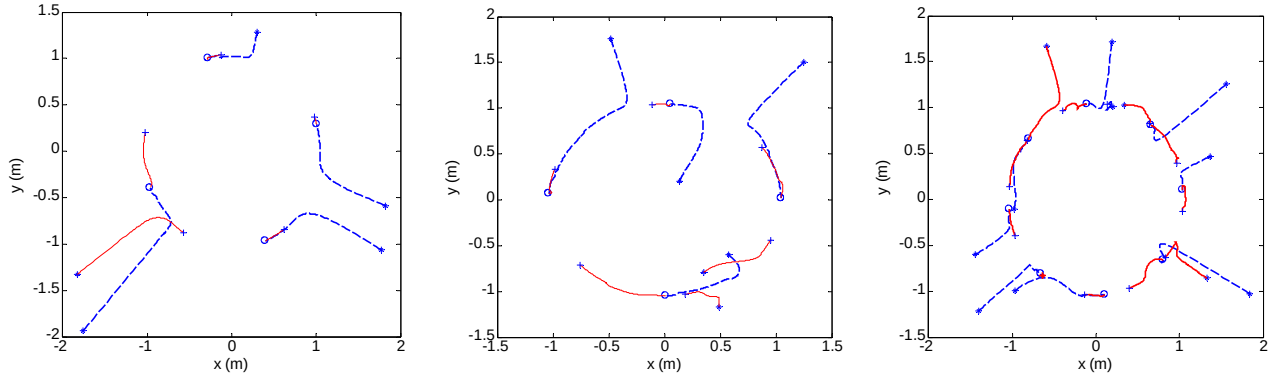
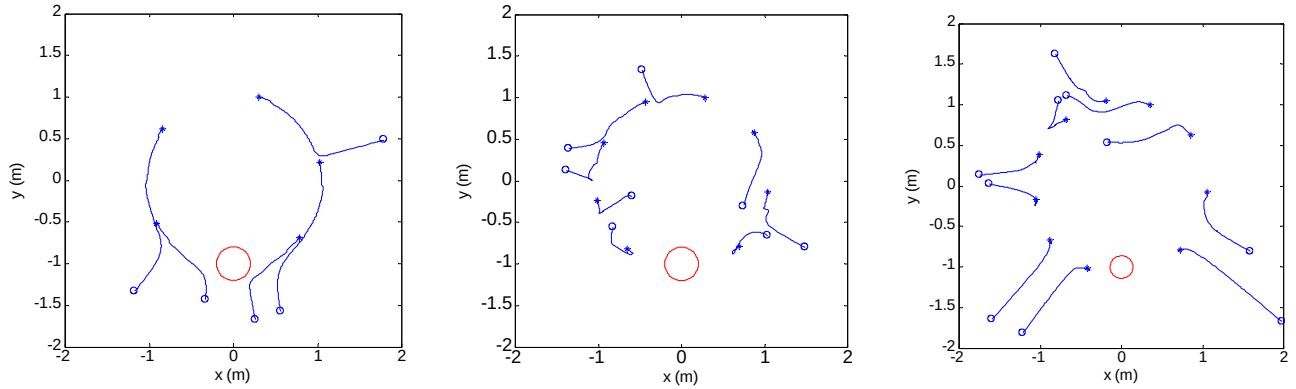


Fig.10. 8 robots are deployed uniformly on a unit square. The convergence time is 0.84 and the total travel distance is 29.03.



(a) 4 robots with 1 newcomer; (b) 4 robots with 2 newcomers; (c) 8 robots with 4 newcomers.

Fig. 11: Trajectories of multi-robots during self-organization with newcomers. The initial positions of the robots are plotted as '*', the intermediate states where the first batch of robots are located are denoted as 'o', and the final states of all the robots are denoted as '+', the dash lines denote the initial deployment trajectories of the first batch of robots and the solid lines the trajectories of all the robots after incorporating newcomers.



(a) 5 robots with one obstacle; (b) 8 robots with one obstacle; (c) 10 robots with one obstacle.

Fig. 12. Trajectories of the robots in the self-adaptation test cases. The big red circle represents one obstacle in the environment. The initial positions of the robots are denoted as 'o', and the final states of robots are denoted as '*'.

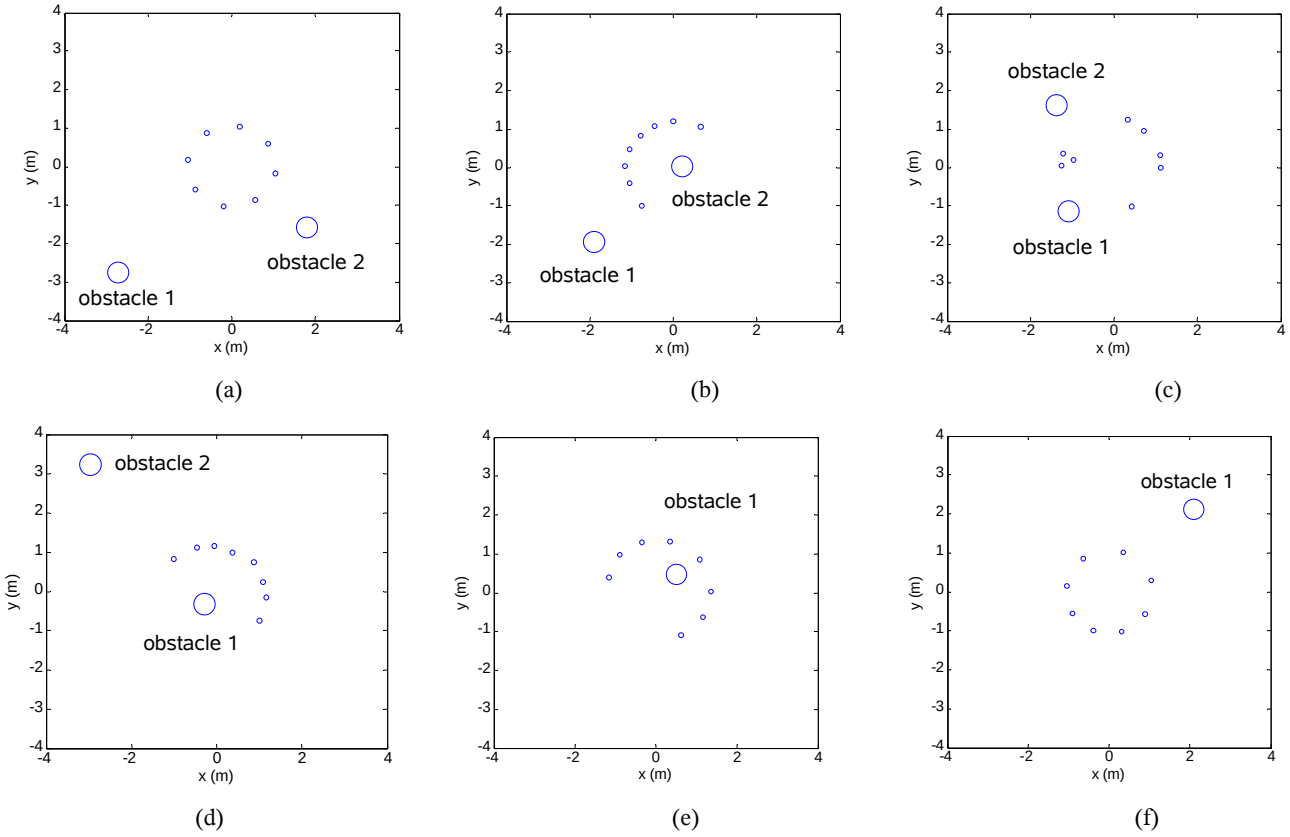


Fig. 13. A set of snapshots showing the behaviors of 8 robots adapting to two moving obstacles. (a) Obstacle 1 and obstacle 2 are moving towards the robots; (b) Robots adapt themselves to avoid obstacle 2; (c) Robots are avoiding both obstacles; (d) Robots are avoiding obstacle 1, while obstacle 2 is leaving away from the simulation environment; (e) Robots are adapting to avoid obstacle 1 while obstacle 2 disappears; (f) Obstacle 1 moves away.

D_i

Table 1. The parameter of three Pareto-optimal solutions and their corresponding performance marked in Fig. 2.

	k	c	b	a	m	total travel distance (meter)	convergence time (second)
(a)	52.2443	77.8411	39.8006	8.5688	4.7916	19.6390	0.3900
(b)	48.5385	78.3474	39.4387	14.8201	8.0243	19.7704	0.2500
(c)	42.8861	99.0034	35.3917	64.8640	42.4166	20.5041	0.0950

Table 2. The performance of the systems for case study 1.

# of robots	total travel distance (meter) (mean std)	convergence time (second) (mean std)	position error (meter) (mean std)
5	13.5603 2.3147	0.1566 0.0257	0.0211 0.0012
10	25.8878 4.3453	0.3060 0.0727	0.0252 0.0021
20	54.2412 8.2531	0.6097 0.1357	0.0304 0.0105

Table 3. The performance of the systems for case study 4.

# of robots	total travel distance (meter) (mean std)	convergence time (second) (mean std)	position error (meter) (mean std)
5	13.8502 2.2528	0.1480 0.0204	0.0205 0.0073
8	22.8615 3.0643	0.2760 0.0626	0.0291 0.0090
10	28.9114 5.6362	0.3368 0.0828	0.0315 0.0112

D_i