Brain Storm Robotics: An Automatic Design Framework for Multi-Robot Systems

Jian Yang, *Member, IEEE*, Yang Shen
Dept. of Computer Science and Engineering
Southern University of Science and Technology (SUSTech)
Shenzhen, China
yangj38@mail.sustech.edu.cn, sheny3@mail.sustech.edu.cn

Yuhui Shi*, *Fellow, IEEE*
Dept. of Computer Science and Engineering
Southern University of Science and Technology (SUSTech)
Shenzhen, China
shiyh@sustech.edu.cn

***Abstract*—Designing the collaborative mechanism is a fundamental problem for the multi-robot systems. It aims to determine the perception, communication, and motion strategies for a single robot to obtain the desired behavior at the system level. Generally, we can use manual design and automatic design, or the combination of the above two approaches for particular system behavior. With the increasing of task complexity and the uncertainty of surroundings, the adaptability and autonomy are hard to achieve with manual design approaches. By using the mechanism of learning or evolution, automatic design can generate sensor configurations, communication parameters, as well as control strategies automatically, which has been widely concerned in recent years. In this paper, the brainstorming method of collaborative problem-solving in human society is introduced into the design of multi-robot systems. This paper proposes an automatic design framework: Brain Storm Robotics (BSR), in which the system architecture, the representation of ideas, and the generation of new ideas are discussed. The effectiveness of the proposed BSR framework is verified by an example of designing an aggregation behavior for a swarm of robots. The results show that the control strategy designed by this framework is more efficient than that designed manually, which has outstanding development prospects. The future researches for the development of this potential framework are also discussed.**

***Index Terms*—Multi-robot Systems, Swarm Robotics, Automatic Design, Fuzzy Control, Brain Storm Optimization**

I. Introduction

Multi-robot systems (MRS) have received much attention due to its enormous potential for applications such as environmental monitoring, collaborative exploration, search and rescue, cooperative manipulation, as well as military defense scenarios [1]. The motivations for developing the multi-robot systems commonly lie in their advantages compared to single-robot systems, including the parallelism for distributed complex tasks, more simplicity of building several resource-bounded robots than a single powerful robot, and also the increasing of robustness through redundancy. There are mainly two types of multi-robot systems: Intentionally Cooperative Systems (ICSs) and Swarm Robotics (SR) [2]. The ICSs generally have several heterogeneous agents with different functions for particular tasks [3]. They interact intentionally to achieve task requirements. While swarm robotics is a particular approach to multi-robot systems, and takes inspiration from the behaviors of social animals or insects, which generally have individuals with low-complexity [4]. SR systems typically are composed of a large number of homogeneous individuals with low complexity (may have a small number of different roles) and finish the tasks through local sensing and limited interactions. The main differences between ICSs and SRs are summarized in Table I.

TABLE I: Differences between ICSs and SRs

	Intentionally Cooperative Systems	Swarm Robotics
No. of Robots	Small Fixed Number	Ranging from a few to Thousands
Control Architecture	Decentralized	Centralized or Decentralized
Heterogeneity	Homogeneous	Heterogeneous
Scalability	High	Low
Prior Information of Env.	Unknown	Known or Unknown

Designing a collaborative mechanism is one of the fundamental problems for multi-robot systems. It aims to determine the individual behaviors and interactions with other members or with the environments so that to achieve the global goal of the system, i.e., to design rules at the individual level, which correspond to a desired behavior at the system level. The design may include aspects of perception, control, communication, computation, and heterogeneities, etc. [5]. Therefore, the challenges for this problem come from the integration of different disciplines such as control theory, artificial intelligence, robotics, optimization, as well as bionics. For ICSs, behavior-based approaches are wildly used to solve the design problems for a specific task. The main idea is to decompose a useful behavior for a particular task into primitive control strategies and interaction rules for individuals. For Swarm Robotics, a common way is based on swarm intelligence, which takes the inspiration from the macroscopic collective behaviors in social animals or insects [6], [7], or inspired by multicellular mechanisms such as morphogen diffusion, gene regulatory

This work is partially supported by National Key R&D Program of China under the Grant No. 2017YFC0804003, National Science Foundation of China under grant number 61761136008, Shenzhen Peacock Plan under Grant No. KQTD2016112514355531, Program for Guangdong Introducing Innovative and Entrepreneurial Teams under grant number 2017ZT07X386, and the Science and Technology Innovation Committee Foundation of Shenzhen under the Grant No. ZDSYS201703031748284, and SUSTech Artificial Intelligence Institute (SAINT). Corresponding Author: Yuhui Shi.

networks, etc. [8].

Generally, the design problem has two main categories: manual design and automatic design [9]. The behavior-based and the direct imitations of swarm intelligence roughly belong to the manual type. This design method usually uses a divide-and-conquer mechanism, in which the global behavior of the system for a particular task is decomposed into a set of behaviors for individuals, and establish a finite state machine (FSM) to determine which action an individual will perform according to the sensor inputs [10]. The techniques which adopt the reinforcement learning strategy [11] or evolutionary robotics approaches can be seen as the automatic design category [12]. Those methods typically use the trial and error mechanism to train or evolve a neural network (NN) controller [13] or a fuzzy controller [14] for each member in a team. There does not exist a straight separation line between these two approaches. Other optimization-based approaches use a developed model as a source of inspiration for the designer. Those models sometimes are derived out from the observations from natural processes, such as the virtual physics models [15], or sometimes comes from other disciplines such as control system models [16]. Then the parameters of the model are refined by different optimization algorithms [17]–[19]. Those types of methods are at the intersection of manually and automatic design, as shown in Fig.1.

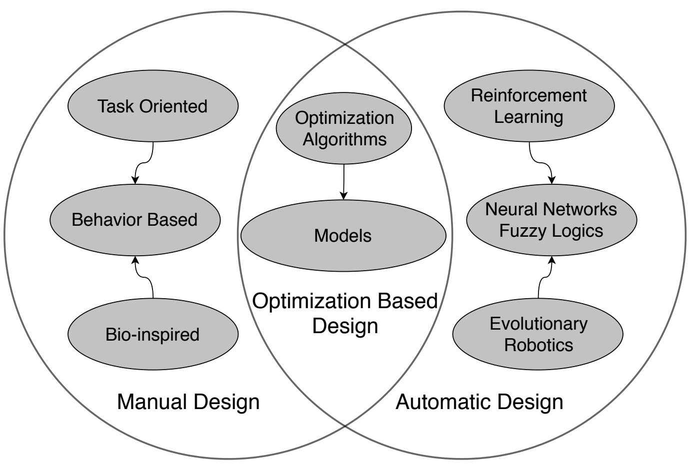


Fig. 1: Catagories of Multi-robot Systems Design

Swarm intelligence algorithms, which are inspired by social animals, were successfully applied in many optimization problems in recent years. Furthermore, due to their capacity for both learning and evolution, they can also be developed as automatic design methods for the multi-robot systems. In this paper, we will propose an automatic design framework, i.e., brain storm robotics (BSR), for multi-robot systems, which is motivated by the problem-solving means of a group of human beings, and is also the basis of Brain Storm Optimization algorithm (BSO) [20], a relatively new algorithm in the field of swarm intelligence. The basic idea of the proposed framework is to let the robots in a team generate "ideas" for control or interaction rules, then to evaluate the results at the system level. A relatively optimal "idea" will be obtained after iterations of generating and evaluating new "ideas".This framework has the characteristics of both optimization and evolution, and is able to increase the capacity of members to generate better new "ideas" by adding learning mechanisms for individuals.

The rest of the paper is organized as follows: Section 2 presents the BSR auto-design framework, including the architecture, the mechanism of idea representation and generation, and the auto-design procedure. Section 3 and Section 4 give a BSR based auto-design example and the results for aggregation behavior of a team of homogeneous robots. Section 5 discusses the potential of the proposed framework and future works. The conclusion is reached in Section 6.

II. Brain Storm Robotics Framework

A brainstorming process generally follows the following steps [20]: 1) get together a group of people with diverse background; 2) generate many ideas; 3) evaluate and pick up better ideas; 4) generate more ideas; 5) pick up better ideas and hopefully a good enough solution can be obtained. In addition, in order to generate ideas with sufficient diversity, four rules of Osborne brainstorming must be considered [21]. By regarding robots as people in the brainstorming process, and the corresponding aspects of being designed as ideas, we will map this procedure to the multi-robot system design problems in the following.

A. Architecture

The proposed architecture is shown in Fig.2, in which each robot is equipped with sensors, actuators, a communicator, a controller, and a planner. The sensors collect the inputs from the surroundings, and the actuators are responsible for the execution of corresponding actions. The controller provides the commands required by the actuators according to the inputs of the sensors and the instructions from the planner. In the proposed framework, the communicator and planner work in two modes: designing mode and working mode. In designing mode, the planner generates "ideas" according to the sensor inputs and actuator outputs (can be based on some learning mechanism). The communicator not only broadcast and receive information from other members in a certain range (which is the standard function in working mode) but also transfer the "ideas" to the upper layer for evaluation and selection. In working mode, the planner module provides corresponding instructions to the controller according to the specific task requirements and the information from sensors, communicators, and actuators. The upper evaluation and selection module only works in the designing mode. According to a specific design objective, the module evaluates the performance of the system based on the "ideas" reported by each planner and generates new "ideas" for the system to execute iteratively, until a relatively optimized "idea" that meets the design requirements is obtained.

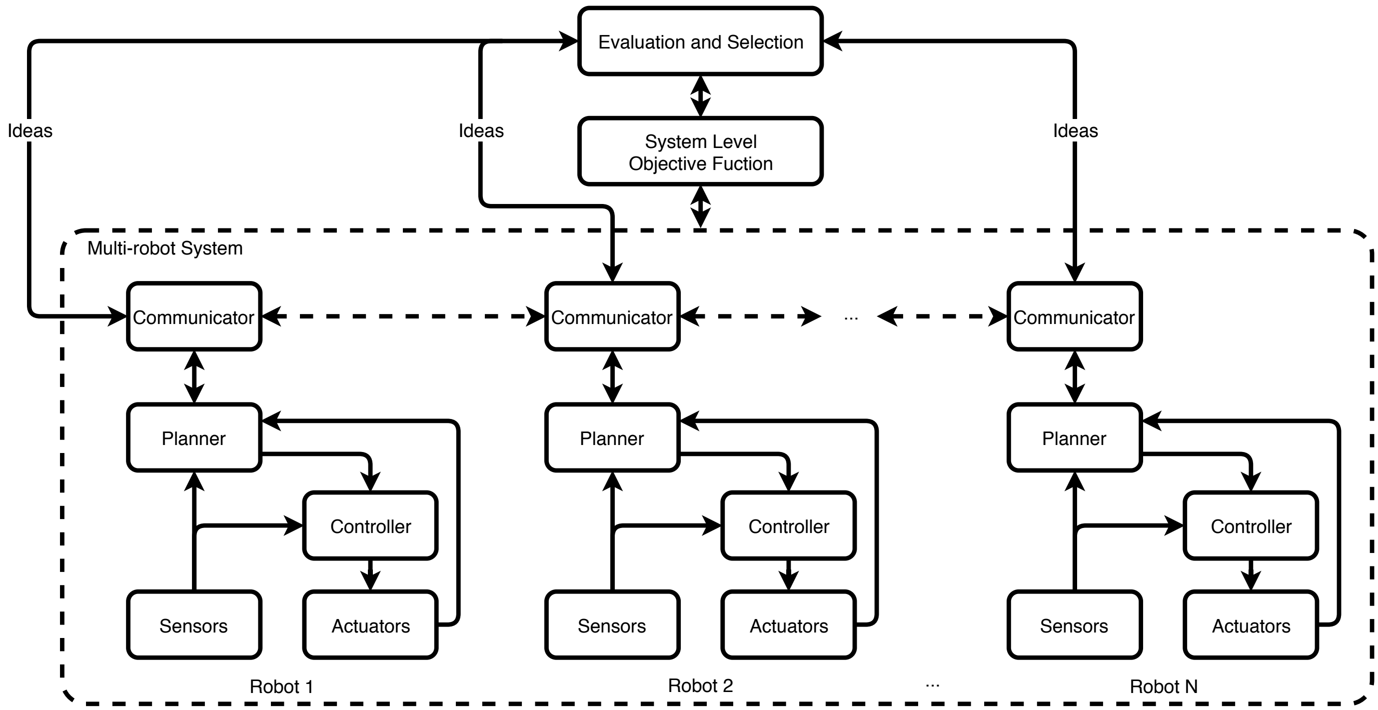


Fig. 2: The Architecture of Brain Storm Robotics

B. Ideas Representation

The representation of ideas may have varied forms according to which design aspect being considered. It may include the sensors configuration, communication range and contents, the controller parameters, as well as the system heterogeneities. The ideas for the controller part will also be different depending on which type of intelligent controllers the designer choose. If the controller is a fuzzy controller, the corresponding ideas may include rule base, number of rules, membership functions for corresponding variables or fuzzification and defuzzification operations [22]. If the controller is a neural network controller, the idea may include the parameters for network structure, connection weights, and activation functions, etc. [23]. The representation of an idea can be visualized by Fig.3.

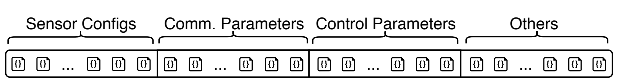


Fig. 3: Idea Representation in BSR

C. New Ideas Generation

Ideas can be generated randomly by each member robot or can be generated according to each one's inputs, outputs, and internal states with a learning strategy. Furthermore, according to the principle of brainstorming, new ideas can not only be proposed by robot members in a group but also be generated from existing ideas by adding some disruption to an existing idea or by combining some existing ideas. Under the consideration of the balance of exploration and exploitation principle in swarm intelligence, here we borrow two predefined probability thresholds from the original BSO-OS algorithm [24], p_r and p_{one} in $[0, 1]$ for new idea generation. The procedure of the generation of new ideas is given in Algorithm 1. $rand_1$ and $rand_2$ are random numbers generated in each iteration in $[0, 1]$, if $rand_1 < p_r$, a new idea will be generated by robots in the group, otherwise will be generated by archived ideas. If $rand_2 < p_{one}$, the new idea will be generated by one robot or one archived idea. Otherwise, the new idea will be generated by multiple randomly selected robots or existing ideas.

Algorithm 1 Procedure of Generate New Ideas

```
if rand_1 < p_r then
    Generate a new idea based on robots:
    if rand_2 < p_one then
        Generate a new idea from one randomly selected robot;
    else
        Generate a new idea from multiple randomly selected robots;
else
    Generate a new idea based on archived ideas:
    if rand_2 < p_one then
        Generate a new idea based on one randomly selected existing idea;
    else
        Generate a new idea based on multiple randomly selected ideas;
```

For new ideas generated by one robot, a randomly selected robot will propose an idea randomly or based on some generation rules, which can be learned from the historical data of its inputs, outputs, and internal states, etc. For the operation of generating a new idea from multiple robots (denotes the total number of ideas as N), the new idea will be formed by taking corresponding parts randomly from the proposed ideas. A combining operation with $N = 2$ is shown in Fig.4. For the situation of generating new ideas based on one archived idea, some disruption will be added to one existing idea to get a new one, as shown in Fig.5. The arrows in Fig.4 and Fig.5 indicate to replace the corresponding randomly determined items in an idea. Besides, if new ideas are generated from multiple existing ideas, the corresponding operation is the same as the mechanism of generating new solutions from multiple robots.

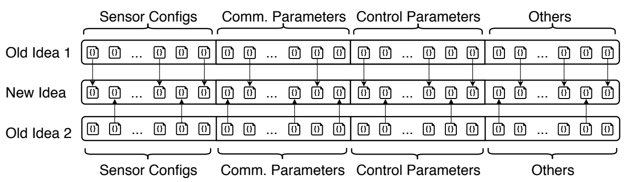


Fig. 4: Generate a New Idea From Two Ideas

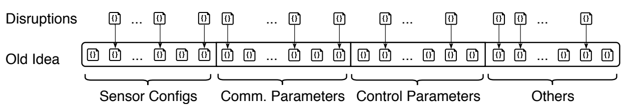


Fig. 5: Generate a New Idea From One Existing Idea

D. *Ideas Evaluation and Selection*

As mentioned above, all ideas will be handled by the evaluation and selection unit for next operations through each one's communicator. The generated ideas from member robots or archived list of ideas will be evaluated and updated iteratively until the relatively acceptable solutions achieved or maximum iteration steps (G) reached. A list with top M better ideas will be kept during the procedure of iterations. For each evaluation, the evaluation and selection unit sends the strategy contained in new ideas to the multi-robot system in turn. Then according to the objective function related to a particular task, with the observation of the whole multi-robot system for a predefined time interval (T), the unit will obtain the corresponding fitness value for each idea. Besides, the evaluation and selection module will select and retain the list of archived better ideas after each round of evaluation.

E. *Automatic Design Procedure with BSR*

The procedure for automatic design with BSR is given in Algorithm 2. First, the parameters for designing will be initialized. Then ideas will be gathered from each robot, after the evaluation of each idea for time T, the top M ideas will be kept in a list according to their fitness values. Afterward, the designing process will enter the iteration process. For each iteration, a new idea will be generated according to Algorithm 1, and the new idea will be evaluated for time T. Then the archived list of ideas will be updated. The iterative process will continue until a good enough solution is obtained, or the maximum number of iterations is reached.

Algorithm 2 Procedure of Auto-Design with BSR

Initialize parameters of p_r, p_{one}, N, M, G, T;
Gathering ideas from robots;
Evaluate each idea for time T;
Keep top M ideas;
while not terminated **do**
    Generate a new idea according to Algorithm 1;
    Evaluate the generated idea for time T;
    Update the top M idea list;
Output the best idea in the archived list;

III. An Automatic Design Example with BSR

For the sake of simplicity, here we give an example of using the proposed BSR framework to design the aggregation behavior of swarm mobile robots automatically. Since the fuzzy control system allows the designer to analyze and interpret the resulting robotic behavior in the context of automatic design techniques, here we assume that each member robot will determine the controller outputs according to the sensor inputs through a fuzzy controller. As mentioned earlier, when using the proposed platform to design a fuzzy controller, the idea may include the corresponding membership function, the strategy of fuzzification and defuzzification, and the rule base as well. As an example, here we only demonstrate how to use the proposed framework to automatically design the fuzzy rule base for the aggregation behavior of a swarm of robots. It can be applied to other aspects of the automatic design mentioned above.

A. *Member Robot*

The member robot configuration is shown in Fig.6, where a differential-driven mobile robot is adopted. The kinematic model of this kind of robot is given in Equation (1) [25].

$$\begin{bmatrix} x(t+\Delta t) \\ y(t+\Delta t) \\ \theta(t+\Delta t) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} + \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

Where (x, y, θ) represent the Cartesian position and heading of the robot, v and ω are the linear and angular velocities in each agent's coordinates (the direction of v is the positive direction of x-axis). Sensors equipped with robots can detect the range and bearing angle of obstacles and other companions in front of the robot within a certain range R.

Simply, the anti-collision and obstacle avoidance operations will be determined according to the preset safety distance d_s. If the range and bearing angle of the nearest obstacle or teammate are d_c and a_c, respectively, the robot's control inputs can be determined by:

$$\begin{cases} \omega = -a_c \\ v = a_0 \frac{d_c}{d_s} \end{cases} \qquad d_c < d_s \tag{2}$$

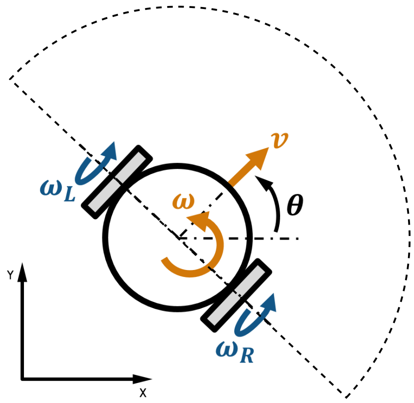


Fig. 6: The Robot Configurations

Where the a_0 is a scale factor, and the v decreases with the decrease of d_c, which avoids the risk of collision.

B. Fuzzy Controller for Self-Organized Aggregation

According to the task requirements, when a member robot is not in the state of collision avoidance, the aggregation behavior will be performed. The aggregation behavior requires the robot to keep a short distance from the surrounding robots as much as possible. Denote the range and bearing angle of all robots around a robot are d_i and a_i respectively, the vector sum of all detected teammates can be expressed as:

$$\begin{cases} |\vec{R}| = ||\sum_i d_i \cos a_i, \sum_i d_i \sin a_i|| - d_s \\ \angle\vec{R} = \arctan(\sum_i d_i \sin a_i, \sum_i d_i \cos a_i) \end{cases} \tag{3}$$

Since the anti-collision operations are performed independently, here the magnitude $|\vec{R}|$ and orientation $\angle\vec{R}$ constitute the two input variables to the aggregation fuzzy controller that regulates the output of the tuning speed ω of the robot. As an example, here we adopt fixed membership functions for the input and output variables [26]. In this example, the proposed BSR is only used to design the rule base of the fuzzy controller automatically.

Suppose the sensor ranges for robot detection are [0 10]m for ranging and [-1.57 1.57]rad for bearing. The input space is partitioned into four trapezoidal fuzzy sets labeled $\{ZO, FR, MF, VF\}$ for the magnitude and seven trapezoidal fuzzy sets labeled $\{VL, ML, LT, ZO, RT, MR, VR\}$ for the orientation of the vector sum of detected teammates, as shown in Fig.7.

The domain of the turning speed [-3 3]rad is partitioned into nine triangular output fuzzy sets labeled $\{VR, MR, RT, ZR, ZO, ZL, LT, ML, VL\}$, which is shown in Fig.8. Here the VR and VL correspond to sharp turns at near maximum speed of 3 rad/s. The terms $\{RT, ZR, ZO, ZL, LT\}$ correspond to relatively smooth turns with $|\omega| \leq 0.5rad$/s.

The aggregation behavior of a swarm of robots is realized by a 2-inputs-1-output standard Mamdani fuzzy controller whose rule base contains 28 rules. A manually designed decision table is provided in Table II.

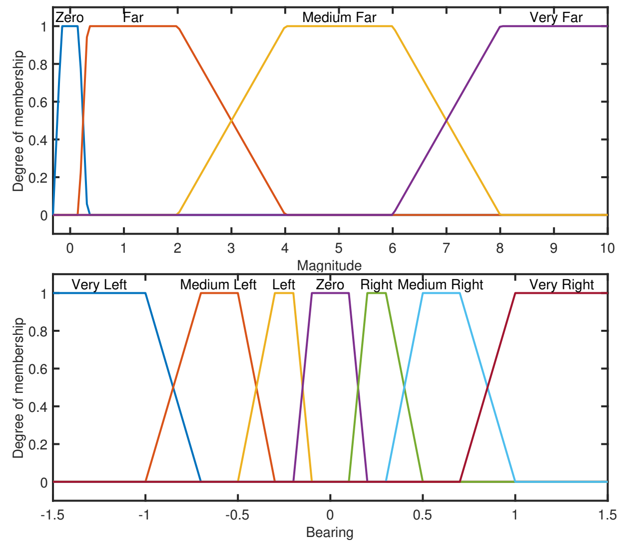


Fig. 7: Membership functions for input variables

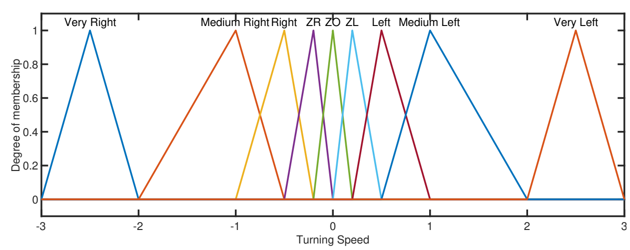


Fig. 8: Membership functions for output turning speed

C. Designing the Rule Base with BSR

Based on the statements and assumptions above, the idea for this example will only encode the rule base with a fixed length, i.e., $\{r_1, r_2, \cdots, r_{28}\}$. The 28 items in the idea can be coded as intgers, i.e., $r_i \in \{1, 2, \cdots, 9\}$ represents possible output fuzzy label $\{VR, MR, RT, ZR, ZO, ZL, LT, ML, VL\}$ correspondingly. According to the procedure described in Algorithm 2, the rule base for aggregation behavior of a swarm of robots can be obtained automatically. Since the desired behavior requires all member robots to gather together as much as possible, the objective function used for system performance evaluation can be written as:

$$\min A_{ch}(R_1, R_2, \cdots, R_K) \tag{4}$$

Where the $A_{ch}(\cdot)$ is the area of the convex hull formed by all the robots in the system, which is depicted in Fig.9.

TABLE II: Manually Designed Decision Table

	ω	$\angle\vec{R}$						
		VL	ML	LT	ZO	RT	MR	VL
$\lvert\vec{R}\rvert$	ZO	ZO	ZO	ZO	ZO	ZO	ZO	ZO
	FR	LT	ZL	ZL	ZO	ZR	ZR	RT
	MF	ML	LT	LT	ZO	RT	RT	MR
	VF	VL	ML	ML	ZO	MR	MR	VR

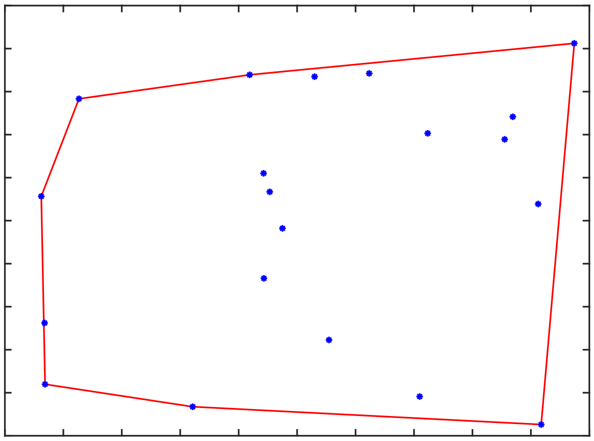


Fig. 9: Convex hull formed by all robots

IV. Results

The simulation is implemented with Mobile Robotics Simulation Toolbox in Matlab 2019b on an iMac with 3.6 GHz Intel Core i9, 8GB DDR4 memory. As depicted in Table III, the number of robots is set to 20, p_r and p_{one} are set to 0.2 and 0.8 respectively. The number of ideas for the operation of generating new ideas based on multiple ideas is 2. The maximum iteration time is 5000 rounds. The time for each idea evaluation is 600 steps.

TABLE III: Simulation Configuratons

No. of Robots	p_r	p_{one}	N	M	G	T
20	0.2	0.8	2	20	5000	600

The sensor range of a single robot is set to 10m with the πrad field-of-view in front, and the safe distance d_s =0.8m. For each evaluation, the robots are distributed randomly in a 20×20m square area. In the beginning, each robot generates an idea randomly. After the evaluation of the initially generated idea, the new idea is generated and evaluated according to the method introduced above. As an example, in this simulation, when a robot generates a new idea, the learning operation of the generation rules has not been added. One of the simulated evaluation processes is visualized in Fig.10.

After 5000 iterations, a relatively optimal solution of the archived ideas is outputed as the final solution, which is given in Table IV. The comparison of control surfaces of the manual and automatic solutions are shown in Fig.11. It can be seen that the biggest difference between the two solutions is that the automatically designed solution significantly reduces the usage of ZO output, which shows that the robot in the solution will be more active in finding the surrounding partners, so as to keep gathering with them.

Furthermore, after another evaluation of these two solutions, the average convex hull area of the manually designed solution is 48.9550. In contrast, that of the automatic design solution is 33.0082, which is significantly lower than the manually

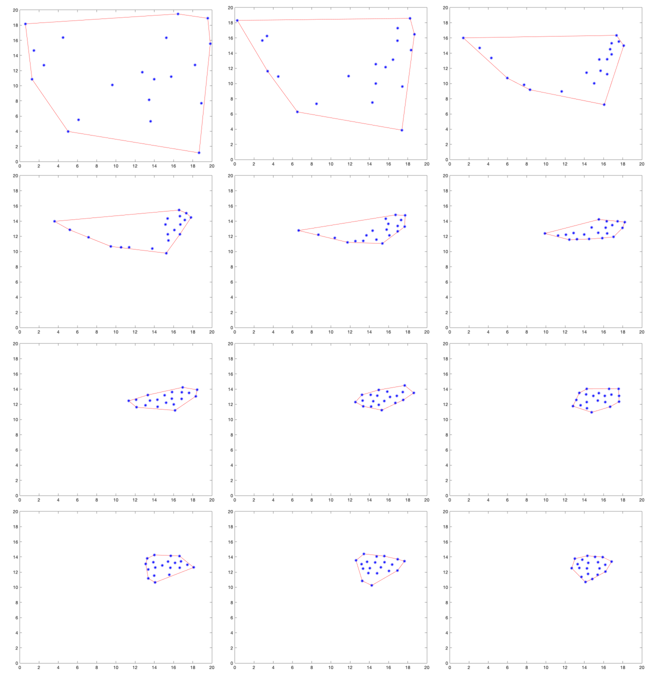


Fig. 10: Visualization for one of the evaluations

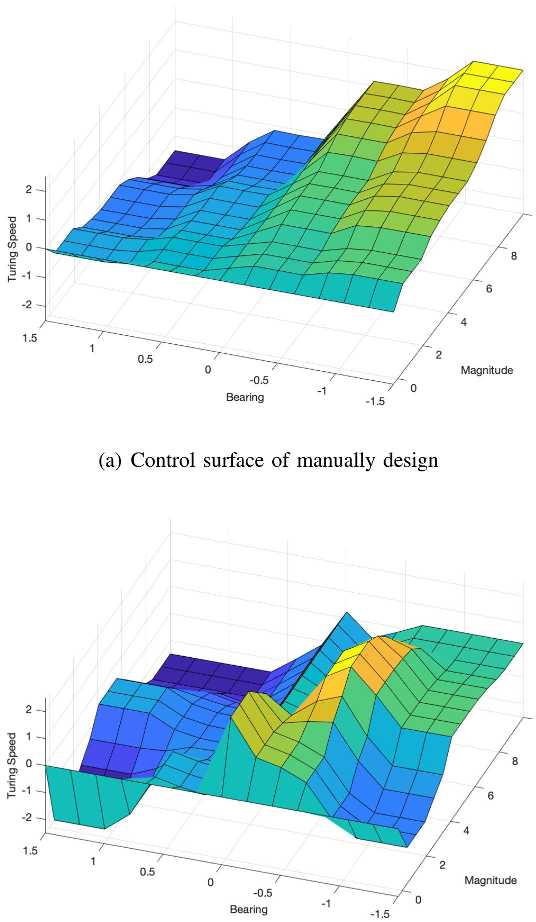
TABLE IV: Automatically Designed Decision Table

	ω	$\angle\vec{R}$						
		VL	ML	LT	ZO	RT	MR	VL
$\|\vec{R}\|$	ZO	VR	RT	ML	ML	MR	VL	LT
	FR	MR	LT	ML	VL	RT	ZR	VR
	MF	LT	VL	ZO	ZL	MR	MR	RT
	VF	ZL	ZL	RT	ZL	MR	VR	VR

designed one. This result further shows that although the solution of automatic design based on BSR framework does not look as "reasonable" as that of manually designed, the performance of task execution is better. Besides, we also know from observation that the collision avoidance operation of the controller designed automatically with the proposed BSR framework is significantly less than that of the controller designed manually. The results above show that the proposed automatic design framework for the multi-robot system is valuable for further development.

V. Discussion and Future Works

Brainstorming, as a kind of collaborative problem-solving method in human social life, has been widely used in solving many kinds of practical problems. One of the characteristics of robots is to substitute human beings to complete some heavy or dangerous tasks. Improving the intelligence and autonomy of robotic systems has always been the focus of academic researches. Naturally, it is of great practical significance to introduce the brainstorming process into the design of multi-robot systems. By employing evolution, optimization, and learning, the BSR framework can automatically determine the decision-making rules of member robots, the interaction with the environment, and other robots through the emergent swarm



(a) Control surface of manually design

(b) Control surface of automatically design

Fig. 11: Comparison of manually and automatically design

intelligence. The above design aspects can be represented into corresponding ideas, and new ideas will be continually generated during the iterative design process. Through the cooperation among members, a final relatively optimal approach will be finally determined. The effectiveness of this method is demonstrated by an example of the aggregation behavior of a swarm of robots.

Our next research will focus on the further development of the framework. On the one hand, we will consider developing the learning ability of individual members, so that when new "ideas" are generated, they are more in line with the robot's state and "experience," that is, the "individual cognition" part of the robot. At the same time, we will also study the mechanism of cooperative learning of multiple robots under the condition of local communication, that is, to develop "social cognition" among members of the framework, so that when new "ideas" are generated, not only based on their own experience but also considering the experience of other members. The introduction of the above learning mechanism will speed up the convergence of the design process and improve the quality of the final solution. On the other hand, we will develop the decentralized distributed structure of the framework. It will make the proposed framework not only be used as the automatic design method of multi-robot systems but also be able to form consistent decision-making through brainstorming mechanism when the actual system is running, facing the complex environment and uncertainty, further improve the ability to solve unknown problems. This method will be developed into a framework that can work for both offline design and online decision making for multi-robot systems.

VI. Conclusion

This paper proposed an automatic design framework for multi-robot systems based on the brainstorming process. This framework aims to introduce the mechanism of human cooperative problem solving into multi-robot cooperation. By representing the different design aspects into some columns of "ideas," new ideas are constantly proposed by robots, or generated by existing ideas. Through a certain number of iterations, a relatively optimal design will be obtained. The effectiveness of this method has been illustrated by an example of automatic designing of the aggregation behavior of a swarm of robots. Simulation results show that the framework can achieve better results than manual design according to the corresponding design objectives. The research of this paper has shown that the proposed BSR framework has strong development potential. On the one hand, we can shorten the design process by adding the mechanism of individual learning and collaborative learning. On the other hand, it can design different aspects such as sensor configuration, communication parameters, and different kinds of controllers, such as fuzzy logic controllers and/or neural network controllers. Furthermore, brainstorming in multi-robot systems can not only be used as an automatic design method but also be used as a collaborative decision-making method for multi-robot systems working in unknown environments by studying its decentralized cooperation mechanism, which has a good development prospect.

References

[1] R. Fierro, L. Chaimowicz, and V. Kumar, "Multi-robot cooperation," in *Autonomous Mobile Robots*. CRC Press, 2018, pp. 417–460.

[2] L. E. Parker, D. Rus, and G. S. Sukhatme, "Multiple mobile robot systems," in *Springer Handbook of Robotics*. Springer, 2016, pp. 1335–1384.

[3] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, p. 29, 2019.

[4] J. Yang, X. Wang, and P. Bauer, "Line and v-shape formation based distributed processing for robotic swarms," *Sensors*, vol. 18, no. 8, p. 2543, 2018.

[5] N. Ayanian, "Dart: Diversity-enhanced autonomy in robot teams," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1329–1337, 2019.

[6] J. Yang, X. Wang, and P. Bauer, "V-shaped formation control for robotic swarms constrained by field of view," *Applied Sciences*, vol. 8, no. 11, p. 2120, 2018.

[7] H. Che, C. Shi, X. Xu, J. Li, and B. Wu, "Research on improved aco algorithm-based multi-robot odor source localization," in *2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, 2018, pp. 1–5.

[8] H. Oh, A. R. Shirazi, C. Sun, and Y. Jin, "Bio-inspired self-organising multi-robot pattern formation: A review," *Robotics and Autonomous Systems*, vol. 91, pp. 83–100, 2017.

[9] V. Trianni, *Evolutionary swarm robotics: evolving self-organising behaviours in groups of autonomous robots*. Springer, 2008, vol. 108.

[10] J. Li and Y. Tan, "A probabilistic finite state machine based strategy for multi-target search using swarm robotics," *Applied Soft Computing*, vol. 77, pp. 467–483, 2019.

[11] G. Sartoretti, Y. Wu, W. Paivine, T. S. Kumar, S. Koenig, and H. Choset, "Distributed reinforcement learning for multi-robot decentralized collective construction," in *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 35–49.

[12] A. E. Eiben, "Grand challenges for evolutionary robotics," *Frontiers in Robotics and AI*, vol. 1, p. 4, 2014.

[13] F. Mukhlish, J. Page, and M. Bain, "Evolutionary-learning framework: improving automatic swarm robotics design," *International Journal of Intelligent Unmanned Systems*, vol. 6, no. 4, pp. 197–215, 2018.

[14] Y. Shi, R. Eberhart, and Y. Chen, "Implementation of evolutionary fuzzy systems," *IEEE Transactions on fuzzy systems*, vol. 7, no. 2, pp. 109–119, 1999.

[15] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[16] T. Keviczky, F. Borrelli, and G. J. Balas, "A study on decentralized receding horizon control for decoupled systems," in *Proceedings of the 2004 American Control Conference*, vol. 6. IEEE, 2004, pp. 4921–4926.

[17] B. Xiao, H. Su, Y. Zhao, and X. Chen, "Ant colony optimisation algorithm-based multi-robot exploration," *International Journal of Modelling, Identification and Control*, vol. 18, no. 1, pp. 41–46, 2013.

[18] J. Yang, X. Wang, and P. Bauer, "Extended pso based collaborative searching for robotic swarms with practical constraints," *IEEE Access*, vol. 7, pp. 76 328–76 341, 2019.

[19] W. Wang, M. Cao, S. Ma, C. Ren, X. Zhu, and H. Lu, "Multi-robot odor source search based on cuckoo search algorithm in ventilated indoor environment," in *2016 12th World Congress on Intelligent Control and Automation (WCICA)*. IEEE, 2016, pp. 1496–1501.

[20] Y. Shi, "An optimization algorithm based on brainstorming process," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 2, no. 4, pp. 35–62, 2011.

[21] A. Osborn, *Applied Imagination-Principles and Procedures of Creative Writing*. Read Books Ltd, 2012.

[22] A. S. Koshiyama, R. Tanscheit, and M. M. Vellasco, "Automatic synthesis of fuzzy systems: An evolutionary overview with a genetic programming perspective," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 2, p. e1251, 2019.

[23] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[24] Y. Shi, "Brain storm optimization algorithm in objective space," in *2015 IEEE Congress on evolutionary computation (CEC)*. IEEE, 2015, pp. 1227–1234.

[25] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions On Robotics And Automation*, vol. 17, no. 6, p. 905, 2001.

[26] F. Hoffmann, "Evolutionary algorithms for fuzzy control system design," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1318–1333, 2001.