Coordinating Cross-robot Resource for Task Allocation with Decentralized Cooperative Genetic Algorithm

1st Yifan Wang
Systems Engineering Research Institute
China State Shipbuilding Corp.,Ltd.
Beijing, China
18601909691@163.com

2nd Jiaxing Mao
Systems Engineering Research Institute
China State Shipbuilding Corp.,Ltd.
Beijing, China
jxingm@163.com

3rd Yu Yang
Systems Engineering Research Institute
China State Shipbuilding Corp.,Ltd.
Beijing, China
yy409046909@hotmail.com

***Abstract*—This paper proposes a decentralized cooperative genetic algorithm (DCGA) to address the cross-robot resource allocation problem in leaderless multi-robot systems (MRS), where each agent is only capable of making local decisions based on local information. Inspired by the fundamental concept of divide-and-conquer in co-evolution mechanisms, the DCGA alternates between two steps: local evolution and global negotiation, within each robot. Our problem formulation enables the algorithm to effectively manage complex cross-robot constraints, which are often challenging in distributed algorithms. Unlike parallelized algorithms that necessitate agents to possess global decision-making capabilities, the DCGA facilitates the seamless integration of agents into multi-agent systems (MAS) by merely incorporating information interaction logic. Experimental results demonstrate that the DCGA is competitive in terms of optimization speed and yields superior solutions compared to centralized genetic algorithms (CGA). Furthermore, it exhibits adaptability to network latency in wireless environments.**

Keywords*—task allocation, multi-robot system, decentralized genetic algorithm

I. Introduction

Multi-robot systems (MRS) have been deployed in numerous real-world scenarios, including disaster response, manufacturing, and reconnaissance, owing to their advantages in efficiency, cost-effectiveness, and safety [1]–[4]. Task allocation represents a fundamental challenge within MRS, where a group of autonomous agents collaborates to efficiently accomplish a set of tasks.

In distributed heterogeneous multi-robot systems, the application of traditional centralized approaches through a single central agent may become impractical or inefficient. This is primarily due to the variability in decision-making logics among robots, which arises from their differing characteristics; for instance, path planning for unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) [5]–[8] can differ significantly, impacting task allocation outcomes. Early research predominantly focused on scheduling with respect to temporal constraints between tasks [10]–[12]. However, in many instances, task allocation must occur in environments with limited resources, where few robots are capable of independently completing a single task.

According to the taxonomy of multi-robot task allocation (MRTA) problems defined in [9], the addressed problem can be categorized into single-task robots, multi-robot tasks, and instantaneous assignment (ST-MR-IA).

Decentralized cooperative approaches have emerged as promising solutions to this challenge. These methods leverage the collective intelligence and autonomy of individual agents to effectively distribute tasks while minimizing the necessity for centralized control and communication. Among various decentralized techniques, genetic algorithms (GA) have demonstrated particular promise due to their capacity to explore extensive solution spaces and adapt to dynamic environments.

The focus of this paper is the development and evaluation of a decentralized cooperative genetic algorithm for task allocation in resource-constrained environments. Unlike traditional centralized methods, the proposed approach enables autonomous agents to collaborate and make local decisions, guided by evolutionary principles inspired by natural selection. By decentralizing the task allocation process, the proposed algorithm seeks to enhance scalability, robustness, and efficiency while accommodating resource limitations.

Key components of the proposed algorithm include individual agent behavior, task representation, genetic operators, and cooperative strategies. Through a decentralized coordination mechanism, agents exchange information, assess potential task assignments, and collaboratively evolve solutions to achieve optimal or near-optimal task allocations given the constraints of available resources.

This paper contributes to the existing literature by presenting a comprehensive framework for decentralized cooperative task allocation utilizing genetic algorithms, specifically tailored for environments with limited resources. The effectiveness of the proposed approach is evaluated through simulations and comparisons with alternative methods, demonstrating its potential applicability in diverse real-world domains that require autonomous multi-agent collaboration.

979-8-3503-8418-5/24/$31.00 ©2024 IEEE

II. Related Work

In the domain of MRS, task allocation poses a critical challenge that involves assigning various tasks to robots in a manner that maximizes overall system efficiency and effectiveness.

Dias and Stentz [15] introduced a decentralized approach for multi-robot exploration tasks, which has demonstrated robustness in dynamic and uncertain environments. Such systems leverage local decision-making, enhancing system robustness by reducing reliance on a central coordinator and improving scalability through the distribution of computation across multiple agents [16].

Auction-based methods offer a dynamic and flexible mechanism for distributing tasks among robots. One seminal work in this area, conducted by Bertsekas [18], describes the auction algorithm for assignment problems, which has been adapted for real-time multi-robot systems by Zlot and Stentz [17]. In their approach, robots bid on tasks based on their capabilities and task requirements, resulting in efficient task allocations that take into account the preferences and operational capabilities of each robot. However, Lagoudakis et al. [19] have shown that the effectiveness of auction-based methods can diminish in highly competitive scenarios, where tasks may attract significantly high bids, leading to inefficiencies.

The complexity of cooperative tasks, in which multiple robots must collaborate to complete assignments, necessitates sophisticated coordination strategies. Service and Adams [20] provide a dynamic programming-based approach to coalition formation, where agent resources are transferable, ensuring the utility of the generated solution. Ayari and Hadouaj [21] proposed a mechanism that operates within a neighborhood agent network, enabling agents to dynamically join new coalitions with higher priority at any time. Wang et al. [22] designed a distributed hedonic coalition formation game method incorporating prioritization and consensus stages to address the task allocation problem for multiple heterogeneous agents. Additionally, [23] examines two centralized algorithms for scheduling robots across tasks and forming suitable coalitions, assuming stochastic travel times between tasks. However, these studies have not extensively considered the dependencies between tasks.

III. Problem Formulation

As an extension of the single-task multi-robot (ST-MR) problem, the studied resource allocation problem can be defined by the following key elements:

-A set of M unmanned robots within a decentralized system, donated as $V = \{v_1, v_2, ..., v_M\}$. Each robot $v_1 \in V$ has a current two-dimension location $P_i = \{x_i, y_i\}$, a load of C types of resources $L_i = \{l_i^1, l_i^2, ..., l_i^C, |l_i^k \in \mathbb{N}\}$, where C is assumed to be a constant, and a maximum traveling distance $D_i \in \mathbb{R}^*$ due to its battery or fuel tank capacity.

-A set of N tasks donated as $T = \{t_1, t_2, ..., t_N\}$. Each task t_j is characterized by its location $P_j = \{x_j, y_j\}$, a reward $W_j \in \mathbb{R}^*$ upon completion, and a resource requirement vector $R_j = \{r_i^1, r_i^2, \ldots, r_i^C | r_i^k \in \mathbb{N}\}$.

-An allocation scheme donated as $X = \{x_{ijk} \in \mathbb{R}^* | 1 \le i \le M, 1 \le j \le N, 1 \le q \le C\}$. For $x_{ijk} \in X$, it represents the amount of resource k that robot v_i allocates to task t_j. The total amount of resource k allocated to task t_j is defined as:

$$\hat{x}_{jk} = \sum_{i=1}^{M} x_{ijk} \tag{1}$$

For ease of description, we consider $C = 1$, thus:

$$\hat{x}_j = \sum_{i=1}^{M} x_{ij} \tag{2}$$

An additionally matrix is defined for convenience as $A = \{\alpha_{ij} | 1 \le i \le M, 1 \le j \le N\}$, where $\alpha_{ij} = 1$ if $x_{ij} > 0$ or 0 otherwise.

-A set of cost function $C = \{C_1, C_2, \ldots, C_M\} : V \times T \to \mathbb{R}^*$, where C_i represents the total cost incurred by robot v_i based on X. Here, we consider the travel distance as the cost c_{ij} associated with robot v_i completing task t_j:

$$c_{ij} = distance(P_i, P_j) * \alpha_{ij} \tag{3}$$

where $distance(P_i, P_j)$ is computed for robot v_i according to its path planning method. If the robot is unable to complete task t_j objectively, $distance(P_i, P_j)$ is set to $\prime$.

Thus C_i is defined as:

$$C_i = \sum_{j=1}^{N} c_{ij} \tag{4}$$

-A profit function F is defined as follows:

$$F = \begin{cases} \sum\limits_{j=1}^{N} \dfrac{\hat{x}_j}{R_j} W_j - \sum\limits_{i=1}^{M} C_i, & if\ \hat{x}_j \le R_j \\ \sum\limits_{j=1}^{N} W_j - \sum\limits_{i=1}^{M} C_i, & else \end{cases} \tag{5}$$

The objective is to find an allocation scheme X that maximize F, subject to the follwing constraints:

$$x_{ij} \ge 0, \forall i \in [1, M], j \in [1, N] \tag{6}$$

$$\sum_{j=1}^{N} x_{ij} \le l_i, \forall i \in [1, M] \tag{7}$$

$$\sum_{j=1}^{N} \alpha_{ij} \le 1, \forall i \in [1, M] \tag{8}$$

These constraints ensure that each robot can be assigned to no more than one task. An allocation scheme X is considered $feasible$ if and only if it satisfies all constraints in (6) to (8).are consistent and known to all robots, allowing them to communicate and determine whether the allocation scheme X is $feasible$.

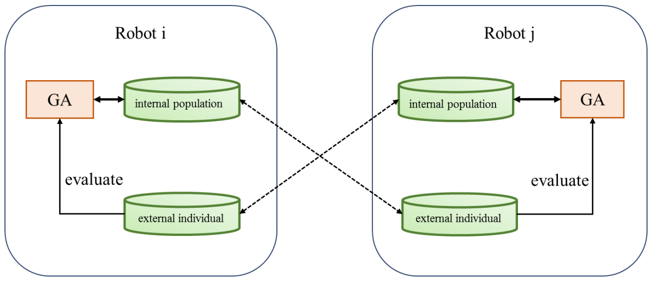


Fig. 1. Structure of DCGA in two-robots instances.

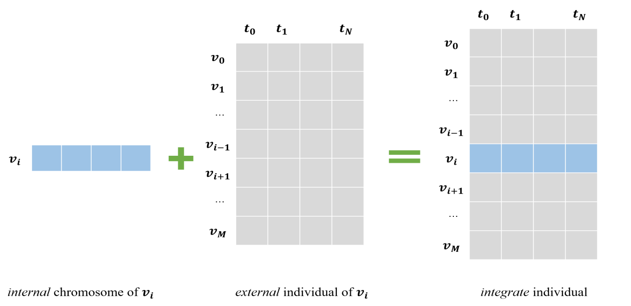


Fig. 2. Relationship among internal chromosome, external individual and integrated individual.

IV. Approaches

The problem is formulated as an integer non-linear programming (INLP) model. Genetic algorithms are a widely used method for solving such problems. These algorithms are probabilistic in nature, generating optimal (or satisfactory) individuals within the population based on the principles of the pattern theorem, rather than relying on gradient information. Consequently, genetic algorithms have lower requirements regarding the objective function and constraints.

For integer programming problems, the space of $feasible$ solutions is limited, allowing for the derivation of a series of optimal (or satisfactory) solutions through the design of appropriate operators [13].

In contrast to parallelized approaches that require all agents to possess global decision-making capabilities, the algorithm proposed in this work is based on the concept of co-evolution. This approach divides the large-scale optimization problem (LSOP) into several smaller or medium-scale subproblems, which are solved independently, followed by a negotiation phase. The structure of the proposed DCGA is illustrated in Fig. 1. Each robot maintains a set of solutions for its respective subproblem, runs a local genetic algorithm (GA) for evolution, and shares information with other robots to construct integrated solutions. The integrated solution may or may not be feasible based on the constraints and is evaluated according to the fitness function (i.e. profit function F). Since all robots share the same profit function F, they are effectively evolving towards a common objective.

A. Encoding

The original problem of solving X is decomposed into the task of finding its linear vector u_i associated with robot r_i. We employ integer encoding for u_i, based on the observation that for any element $x_{ij} \in u_i$, x_{ij} is a bounded non-negative integer. This encoding method utilizes arrays of integers to represent chromosomes, where the integer value at each position (gene) corresponds to a specific component of the solution.

Definition 1: For each robot r_i, its internal population $\mathcal{S}_i$ consists of K chromosomes, where K is assumed to be a constant. Each internal chromosome $s_i \in \mathcal{S}_i$ represents an integer encoding of u_i. Any chromosome s' originating from another robot is considered an external chromosome for robot r_i.

Definition 2: An Integrate individual $\mathcal{A}$ is represented as a two-dimensional matrix, composed of one internal chromosome from each robot. These chromosomes are arranged in a row-wise manner according to the serial numbers of the robots.

Defination 3: An external individual $\mathcal{E}$ of robot r_i is represented as a two-dimensional matrix that excludes the chromosome corresponding to u_i from $\mathcal{A}$.

Fig. 2 illustrates the relationship among the internal chromosome, external individual, and integrated individual. In each robot, the genetic algorithm (GA) operates solely on the internal population.

The external individual is received from other robots and combined locally, with its elements initially set to zero.

B. Algorithm Framework

The pseudocode of DCGA is presented in Algrithm 1. The execution flow is governed by two layers of iterations: the outer layer facilitates communication and negotiation among the robots, while the inner layer manages local evolution. The fitness function, denoted as $Fitness()$, is computed as shown in (5).

1) Self-evolution operators: Crossover and mutation are fundamental operations used to generate new offspring chromosomes (or solutions). These operations enhance the diversity within the population and facilitate the exploration of the solution space.

Self-Crossover. We employ a single-point self-crossover operator that operates on a single parent rather than on multiple parents, given that there is at most one non-zero element in a $feasible$ solution s due to the constraints outlined in (8).

For instance, consider two feasible chromosomes $s_1 = [0, 2, 0, 0]$ and $s_2 = [0, 0, 5, 0]$. If the crossover point is selected as index 2, the general method would produce two offspring: $ch_1 = [0, 2, 5, 0]$ and $ch_2 = [0, 0, 0, 0]$. Here, ch_1 conflicts with the constraints in (8), and ch_2 represents an initialization value. In contrast, the self-crossover approach generates two feasible offspring: $ch_1 = [0, 0, 2, 0]$ and $ch_2 = [5, 0, 0, 0]$, which have a higher probability of being $feasible$. Furthermore, if the crossover point is at index 1 or 3, the offspring produced by the general method would remain identical to the parents. Thus, in this context, crossover on parents is inefficient.

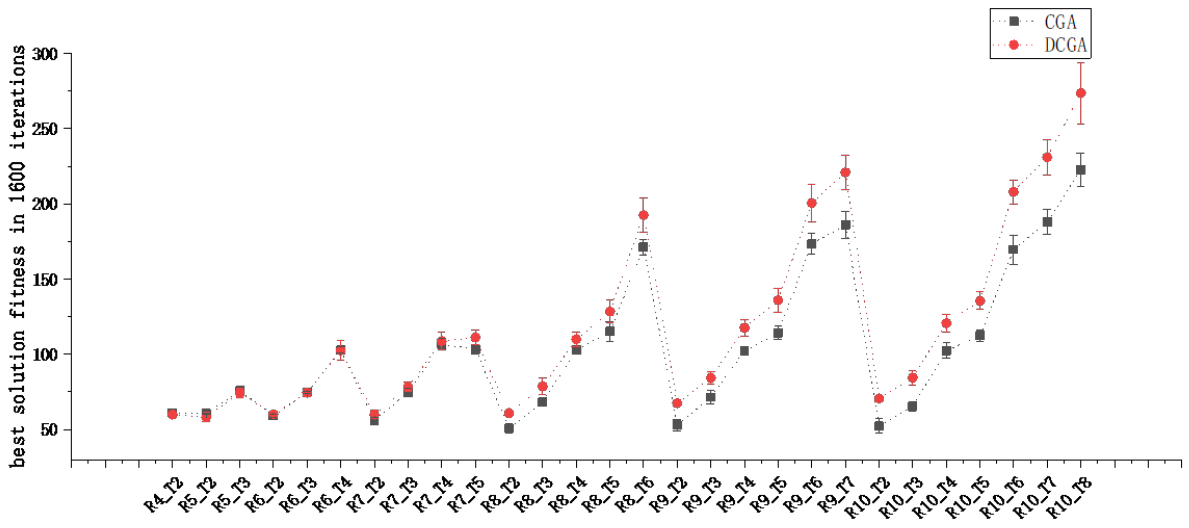


Fig. 3. Comparison of optimization capabilities over 1600 iterations. The horizontal axis of the graph represents various instances, each distinguished by the number of robots (indicated after the letter R) and the number of tasks (indicated after the letter T).

Mutation. The mutation operator used for integer encoding in this study is boundary mutation, which involves perturbing the value of a gene within its predefined boundaries. The gene selected for mutation is either the only non-zero value in a chromosome or a randomly chosen position if all genes are zero.

2) Share internal chromosomes: Following each local evolution, each robot selects p internal chromosomes $\mathcal{S}^p$ from its internal population $\mathcal{S}$, transmits them to all neighboring robots, and simultaneously receives similar messages. In this study, we employ a randomized strategy for the selection of $\mathcal{S}^p$, whereby each chromosome has an equal probability of being included in $\mathcal{S}^p$. Although this strategy may appear to diminish the efficiency of the optimization process, it is analogous to the Metropolis criterion [14] utilized in simulated annealing algorithms. This approach facilitates flexible exploration of the search space, thereby increasing the likelihood of identifying an optimal solution to the problem.

3) Co-evolution operators: In DCGA, we have designed two co-evolution operators known as concession and consensus..

Consession. Each robot's evolution is guided by the same fitness function defined in (5), which leads them to allocate resources to their preferred tasks as much as possible. Consequently, the integrated individuals are likely to become infeasible due to the constraints outlined in (7). The concession operator aims to enhance the effectiveness of the search process by reducing the value of the non-zero genes in the internal chromosome to satisfy the constraints specified in (7).

Consensus. Self-evolution is a stochastic search process, resulting in variability among the integrated individuals across different robots. The consensus operator first shares the best local integrated individual $\mathcal{A}_l^b$ with other robots. If any of the integrated individuals $\mathcal{A}_l'$ received from neighboring robots exhibit greater fitness than $\mathcal{A}_l^b$, the robot r_i will incorporate the internal chromosome from $\mathcal{A}_l'$ its own population $\mathcal{S}_i$, and designate $\mathcal{E}$ of r_i as the external individual in $\mathcal{A}_l'$(i.e., as indicated by the $Update$ function in Algrithm 1).

V. Experiments

In this section, we present the experimental results of DCGA in comparison to a CGA across various problem instances of differing sizes. In all simulations, both the robots and tasks are randomly positioned within a uniform distribution over a 100-meter by 100-meter area. The resources allocated to each robot, denoted as L are randomly generated within the range of [0, 50], while the task rewards, denoted as W, are randomly generated within the range of [100, 200].All statistical data were collected over 20 independent runs. The hyperparameters utilized in the experiments are summarized in Table I.

TABLE I
HYPERPARAMETERS IN EXPERIMENTS

Table	Hyper-parameter Name	
Algorithms	***CGA***	***DCGA***
Initial population size Q	16	16
local evolution times e_l	4	-
communication times c_t	400	-
Total evolution times E_t	-	1600
chromosome number shared p	2	-
consensus frequency c_f	1	-
probability for crossover pc	0.8	0.8
probability for mutation pm	0.8	0.8

A. Converge Speed and Optimization Capacity

In this experiment, we investigate the convergence speed and optimization performance across 28 varying instances characterized by different numbers of robots and tasks.

Fig. 3 illustrates the optimization capacity of the two algorithms. It is evident that when the number of robots is small (fewer than 6 in this experiment), the optimization performance of the CGA is slightly superior to that of DCGA.

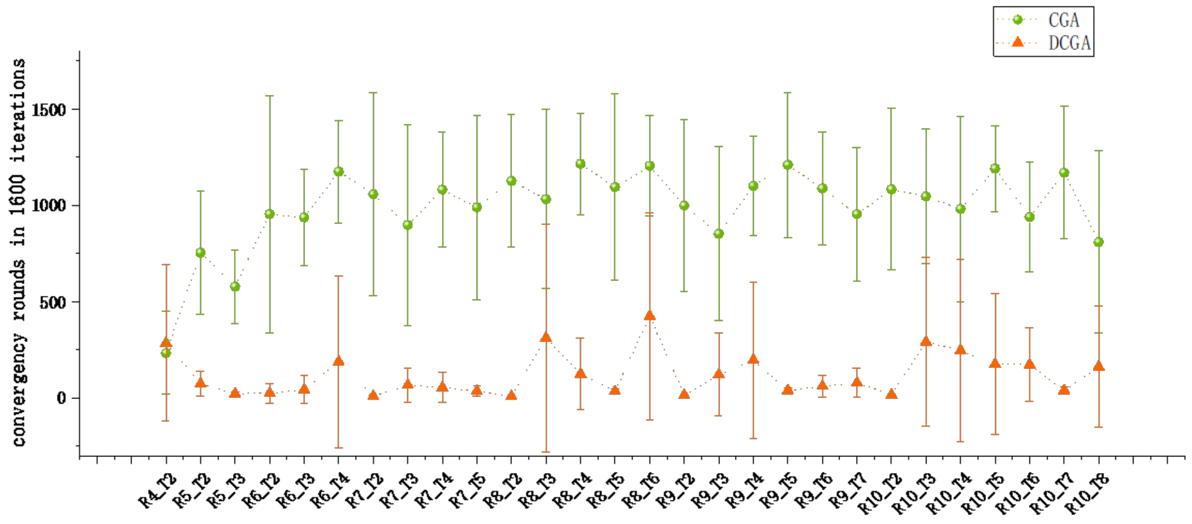
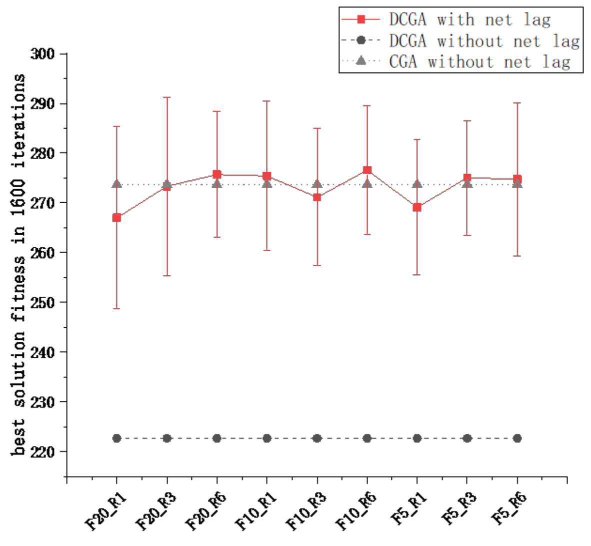
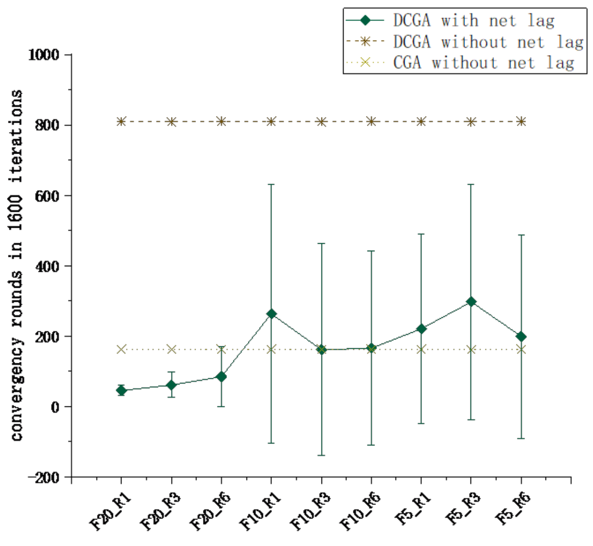


Fig. 4. Comparison of convergency speed over 1600 iterations.



(a) Comparison of Optimization Capabilities.



(b) Comparison of Convergency Speed.

Fig. 5. Performance with or without net lags over 1600 iterations. The horizontal axis of each label represents instances of various communication failures, which are distinguished by the number of offline robots (indicated by the letter R) and the frequency of their offline status (indicated by the letter F).

However, as the number of robots increases, the optimization capability of the DCGA gradually surpasses that of the CGA, with the performance gap becoming increasingly significant. When the number of robots remains constant (for example, 8, 9, or 10), the advantages of the DCGA become more pronounced as the number of tasks increases, indicating a rise in problem complexity.

Fig. 4 displays the convergence speed for the aforementioned 28 instances. We observe that the DCGA demonstrates a clear advantage over the CGA, with the exception of the simplest instance (i.e., 4 robots and 2 tasks).

B. Robustness in the Presence of Communication Failures

Our primary focus is on the performance of the DCGA in the presence of network delays. In MRS, particularly in wireless communication environments, network delays and disconnections are prevalent. Consequently, the performance of this algorithm under such conditions is critical for assessing the robustness of the system.

Network delays impact the consensus operation; therefore, we simulate network disconnections by disabling the consensus operation for one robot during a single iteration. For this experiment, we selected an instance involving 10 robots

Algorithm 1 Decentralized Cooperative Genetic Algorithm

Input: robot list V, task list T, init population size Q, communication times c_t, local evolution times e_l, chromosome number shared in each communication p, and consensus frequency c_f.

Output: resource allocation scheme X.

```
1: Initialize a $Q \times N$ matrix $\mathcal{S}$ with 0
2: Initialize a $(M-1) \times N$ matrix $\mathcal{E}$ with 0
3: Initialize an array $fit$ of size $Q$ with 0
4: for $i = 0 \to c_t$ do
5:     $fit \leftarrow Fitness(\mathcal{S}, \mathcal{E})$
6:     for $j = 0 \to e_l$ do
7:         $\mathcal{S}_{children} \leftarrow Cross(\mathcal{S})$
8:         $\mathcal{S}_{children} \leftarrow Mutate(\mathcal{S}_{children})$
9:         $\mathcal{S}_{children} \leftarrow Consecion(\mathcal{S}_{children}, \mathcal{E})$
10:        $fit_{ch} \leftarrow Fitness(\mathcal{S}_{children}, \mathcal{E})$
11:        $\mathcal{S} \leftarrow Merge(\mathcal{S}, \mathcal{S}_{children})$
12:        $fit \leftarrow Merge(fit, fit_{ch})$
13:        $\mathcal{S} \leftarrow \mathcal{S}[0:Q]$
14:        $fit \leftarrow fit[0:Q]$
15:    end for
16:    if $i\%c_f = 0$ then
17:        $Send(\mathcal{S}, p)$
18:        $\mathcal{E}^{All}_{temp} \leftarrow Receive()$
19:        let $fit_{\mathcal{A}}$ be a 0-array of size $Q \times p^(M-1)$
20:        for $k = 0 \to Q \times p^(M-1)$ do
21:            $fit_{\mathcal{A}}[k] \leftarrow Fitness(\mathcal{S}, \mathcal{E}^{All}_{temp}[k])$
22:        end for
23:        $Sort(Dict(\mathcal{E}^{All}_{temp}, fit_{\mathcal{A}}))$
24:        $\mathcal{E} \leftarrow \mathcal{E}^{All}_{temp}[0]$
25:        $fit \leftarrow Fitness(\mathcal{S}, \mathcal{E})$
26:        $Send(\mathcal{S}[0], \mathcal{E}, fit[0])$
27:        $s_{temp}, \mathcal{E}_{temp}, fit_{temp} \leftarrow Receive()$
28:        $\mathcal{S}, \mathcal{E}, fit \leftarrow Update(s_{temp}, \mathcal{E}_{temp}, fit_{temp})$
29:    end if
30: end for
31: $X \leftarrow \mathcal{S}[0]$
32: return $X$
```

and 8 tasks. The results are presented in Fig 5. Comparatively, our algorithm continues to function effectively under network latency conditions. While the convergence speed of the algorithm is influenced by network delays, its optimization capability remains largely unaffected.

VI. Conclusion

In this paper, we proposed a DCGA to address the cross-robot resource allocation problem in leaderless MRS, which can be classified as ST-MR problems. Unlike parallelized algorithms that necessitate global decision-making capabilities for each agent, the DCGA enables agents to make local decisions and communicate using a standardized message format to facilitate negotiation. This approach enhances compatibility among different agents. Experimental results demonstrated that the DCGA is competitive in both optimization speed and quality compared to the CGA, even in the presence of communication failures. This highlighted its potential for real-world applications across various domains that require autonomous multi-agent collaboration. However, it is important to note that the distributed architecture introduces significant communication overhead, which was not addressed in this study. Future work will focus on adapting our algorithm for large-scale MRS, where the implications of communication overhead become increasingly critical.

References

[1] B. Doroodgar, M. Ficocelli, B. Mobedi and G. Nejat, "The search for survivors: Cooperative human-robot interaction in search and rescue environments using semi-autonomous robots", Proc. IEEE Int. Conf. Robot. Automat., pp. 2858-2863, May 2010.

[2] B. Rabta, C. Wankmüller and G. Reiner, "A drone fleet model for last-mile distribution in disaster relief operations", Int. J. Disaster Risk Reduction, vol. 28, pp. 107-112, Jun. 2018.

[3] H. Shen, L. Pan and J. Qian, "Research on large-scale additive manufacturing based on multi-robot collaboration technology", Addit. Manuf, vol. 30, pp. 100906, 2019.

[4] M. Quann, L. Ojeda, W. Smith, D. Rizzo, M. Castanier and K. Barton, "An energy-efficient method for multi-robot reconnaissance in an unknown environment", Proc. Amer. Control Conf., pp. 2279-2284, 2017.

[5] X. Liu, R. Zhu, "An improved market-based auction algorithm for UAVs task assignment problem",Proc. Int. Conf. Information Technology and Software Engineering, pp. 649-659, Nov. 2012.

[6] H. A. Kurdi, E. Aloboud, M. Alalwan, S. Alhassan, E. Alotaibi, G. Bautista, et al., "Autonomous task allocation for multi-UAV systems based on the locust elastic behavior", Appl. Soft Comput., vol. 71, pp. 110-126, Oct. 2018.

[7] G. Zhang, R.R. Selmic, C.A. Duncan and J. Kanno, "Multi-UGV multi-destination navigation in coordinate-free and localization-free wireless sensor and actuator networks", Proc. 52nd IEEE Conference on Decision and Control, pp. 4134-4139, Dec. 2013.

[8] L. F. Tiotsop, A. Servetti and E. Masala, "An integer linear programming model for efficient scheduling of UGV tasks in precision agriculture under human supervision", Comput. Oper. Res., vol. 114, Feb. 2020.

[9] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems", Int. J. Robot. Res., vol. 23, no. 9, pp. 939-954, Sep. 2004.

[10] S. Choudhury et al., "Dynamic multi-robot task allocation under uncertainty and temporal constraints", Autonomous Robots, vol. 46, pp. 231-247, 2022.

[11] E. Bischoff, F. Meyer, J. Inga and S. Hohmann, "Multi-robot task allocation and scheduling considering cooperative tasks and precedence constraints", 2020 IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 3949-3956, 2020.

[12] H. Wang, W. Chen and J. Wang, "Coupled task scheduling for heterogeneous multi-robot system of two robot types performing complex-schedule order fulfillment tasks", Robot. Auton. Syst., vol. 131, Sep. 2020.

[13] K. Nieminen, S. Ruuth and I. Maros, "Genetic algorithm for finding a good first integer solution for MILP, Technical report, Department of Computing, Imperial College, 2003.

[14] D. Bertsimas et al., "Simulated annealing", Statist. Sci., vol. 8, no. 1, pp. 10-15, 1993.

[15] E. Jones, M. Dias and A. Stentz, "Time-extended multi-robot coordination for domains with intra-path constraints", J. Auton. Robots, vol. 30, no. 1, pp. 41-56, Jan. 2011.

[16] L. E. Parker and F. Tang, "Building multi-robot coalitions through automated task solution synthesis", Proceedings of IEEE, vol. 94, no. 7, pp. 1289-1305, 2006.

[17] R.M. Zlot, "An auction-based approach to complex task allocation for multirobot teams", Ph.D. Thesis, 2006.

[18] D. P. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem", Annals of Operations Research, vol. 14, no. 1, pp. 105-123, Dec. 1988.

[19] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, et al., “Auction-based multi-robot routing”, Proc. Robot.: Sci. Syst., pp. 343-350, 2005.
[20] T. Service and J. Adams, “Coalition formation for task allocation: Theory and algorithms”, J. Auton. Agents Multi-Agent Syst., vol. 22, pp. 225-248, 2011.
[21] G. K. Ayari E and Hadouaj S, “A dynamic decentralised coalition formation approach for task allocation under tasks priority constraints”, ICAR, pp. 250-255, 2017.
[22] L. Wang, T. Qiu, Z. Pu, J. Yi, J. Zhu, and W. Yuan, “Hedonic Coalition Formation for Distributed Task Allocation in Heterogeneous Multi-agent System”, Int. J. Control Autom. Syst., vol. 22, no. 4, pp. 1212–1224, Feb. 2024.
[23] A. Aswale and C. Pinciroli, “Heterogeneous Coalition Formation and Scheduling with Multi-Skilled Robots”, June 2023, [online] Available: https://arxiv.org/abs/2306.11936v1.