Comparison of Distributed Task Allocation Algorithms Considering Non-ideal Communication Factors for Multi-UAV Collaborative Visit Missions

Yan Cao[1], Teng Long[1], Jingliang Sun[1], Zhu Wang[2], Guangtong Xu[3]

***Abstract*—This letter comprehensively investigates the performance of six state-of-art distributed task allocation algorithms (i.e., CBAA, CBBA, HIPC, PI, DHBA, and DGA) subject to non-ideal communication factors. The package loss, bit error, and time delay factors are considered in the distributed task allocation process. The performance of the algorithms for multi-UAV collaborative visit missions is compared under pre-allocation and dynamic allocation scenarios. The synchronous and asynchronous communication modes are separately utilized in different allocation scenarios for analyzing the effects of non-ideal communication factors. Comparison results show that bit error factors cause conflicted allocations. For the pre-allocation scenario, CBBA outperforms the competitors in terms of reliability, communication overhead, and efficiency. For the dynamic scenario, CBBA performs best optimality, while DHBA exhibits better reliability and lower overhead in harsh communication conditions.**

***Index Terms*—Distributed Robot Systems, Task Planning, Networked Robots**

I. Introduction

IN recent years, the cooperative applications of multiple Unmanned Aerial Vehicles (multi-UAV), such as surveillance and reconnaissance, search and rescue, and alternative missions [1], have been widely developed. As a critical technology for performing cooperative missions, task allocation aims to assign the matching pairs between UAVs and targets reasonably to accomplish the expected objectives (minimum range or makespan). Generally, task allocation algorithms can be classified into centralized and distributed categories. The centralized algorithms depend on the central node to generate the allocation plan. As the dimension of the allocation problem grows, the time consumed to reach a feasible solution dramatically increases, which seriously limits the online applications of multi-UAV. The distributed algorithms reach a consensus through exchanging information between UAVs based on a reliable communication link. Therefore, distributed algorithms exhibit better scalability compared with centralized ones. However, in the real world, non-ideal communication factors (e.g., package loss, bit error, and time delay) inevitably exist, and degrade the performance of distributed task allocation algorithms.

Unfortunately, few studies on distributed task allocation algorithms under non-ideal communication have been reported. A systematic comparison of package loss's effects on several distributed task allocation algorithms was provided in [2], and the existing trade-offs are shown between comparing algorithms under different communication conditions. Nevertheless, only package loss factors are considered in [2], which limits the sufficient discussion of algorithm performance. In this letter, more non-ideal communication factors, including package loss, bit error, and time delay, are considered to compare the performance of existing algorithms. To further evaluate allocation algorithms, four metrics, i.e., optimality, communication overhead, efficiency, and reliability, are chosen in the comparison. The involved algorithms include one deterministic integer programming method, i.e., Decentralized Hungarian Based Algorithm (DHBA) [3], one heuristic intelligence optimization method, i.e., Decentralized Genetic Algorithm (DGA) [4], and four auction-based methods, i.e., Consensus Based Auction Algorithm (CBAA) [5], Consensus Based Bundle Algorithm (CBBA) [6], Hybrid Information Plan Consensus (HIPC) [7], and Performance Impact (PI) [8]. The main contributions are listed as follows:

1) It is first observed that bit error factors cause conflicts in the distributed allocation with both pre-allocation and dynamic allocation scenarios.
2) Under synchronous communication, package loss and bit error factors significantly affect the algorithm runtime compared with time delay factors in harsh communication conditions. The asynchronous mode can alleviate the time delay's effects and is suitable for dynamic environments.
3) In the pre-allocation, CBBA exhibits better performance in terms of reliability, communication overhead, and efficiency. For the dynamic allocation, CBBA has the advantage of optimality for minimum path length, while DHBA generates fewer conflicts and requires less communication overhead in harsh environments.

Manuscript received: March 10, 2023; Revised May 30, 2023; Accepted July 4, 2023.

This paper was recommended for publication by Editor Jingang Yi upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Beijing Natural Science Foundation (No. 4214077) and National Natural Science Foundation of China (No. 62003036, 62203256).
(Corresponding author: Jingliang Sun)

[1]Yan Cao, Teng Long, and Jingliang Sun are with the School of Aerospace Engineering, Beijing Institute of Technology, Beijing, 100081 China (e-mail: caoyanbit@163.com; tenglong@bit.edu.cn; sunjingliangac@163.com).

[2]Zhu Wang is with the Department of Automation, North China Electric Power University, Baoding, 071003 China (e-mail: wangzhubit@163.com).

[3]Guangtong Xu is with the Huzhou Institute, Zhejiang University, Huzhou, 313000 China (e-mail: guangtong_xu@163.com).

Digital Object Identifier (DOI): see top of this page.

II. Related Work

The auction-based approach is a major aspect of task allocation methods. The first auction algorithm based on a shared memory was proposed in [9] to address the SR-ST (Single-Robot Tasks vs. Single-Task Robots) [10] problem. Zavlanos et al. [11] extended the auction algorithm to be fully distributed. Choi et al. [5] presented CBAA and used the concept of maximum consensus to distribute auctions over the system. Generally, the SR-ST problem with no dependencies between tasks can be solved in polynomial time complexity. Unfortunately, in multi-UAV practical applications, a coupling dependence between tasks, like In-schedule Dependencies (ID) [10], usually exists and makes task allocation an NP-hard problem. To tackle this issue, CBBA was designed in [5] to solve the SR-ST-ID problem and assigns a task bundle to each agent. Johnson et al. [6] proposed the asynchronous version of CBBA to improve computation efficiency. HIPC [7] and PI [8] realized consensus similarly to CBBA. The differences are that HIPC utilizes global situational awareness rather than greedy bundle construction to enable a higher degree of coordination, and PI defines a new concept of significance instead of bid value and aims to optimize the global objective function. The optimization-based approaches, including deterministic integer programming and stochastic swarm intelligence optimization, are alternative task allocation methods. DHBA [3] is a typical distributed deterministic optimization approach based on the Hungarian algorithm. Representative swarm intelligence approaches include decentralized GA (DGA) [4] and the stochastic Ant-colony optimization algorithm [12].

Most of the existing literature sets different network topologies and simulates the message transmission delay to compare the performance of distributed task allocation algorithms. However, it is challenging to reflect the impact of package loss and bit error on the allocation results during message propagation. Han et al. [13] described the data channel quality with the Signal to Noise Ratio (SNR) threshold when designing the control strategy for guiding UAVs, in which the data channel is reliable only if the SNR exceeds the threshold. Subsequently, Nayak et al. [2] referred to the channel quality concept and proposed the package loss model with Rayleigh fading. Based on [2], Carrillo et al. [14] presented a metareasoning approach that enables multi-UAV to select algorithms adaptively according to the communication quality. To tackle the special issue when communication is very low, Bapat et al. [15] presented two Playbook task allocation algorithms, to generate the same allocation on each UAV by running a deterministic task allocation process. However, the above studies ignored the impacts of bit error and time delay during the message transmission. Kopeikin et al. [16] studied the CBBA algorithm to allocate tasks for UAVs considering Bit Error Rate (BER). Nevertheless, it focused on designing an inter-UAV routing protocol to achieve BER below the threshold instead of analyzing the influence of the bit error on the allocation. CA-CBBA [17] is recently raised with co-designing multiagent algorithms and the communication protocol to deal with the message collision and bandwidth limitation constraints. Li et al. [18] constructed an ad-hoc network simulation platform by integrating the communication protocol stack of the upper layer network and the physical transmission model of the bottom layer, and discussed the performance of PI with synchronous communication.

At present, quite few studies analyze the performance of distributed task allocation algorithms considering package loss, bit error, and time delay simultaneously. Therefore, inspired by [2], this letter discusses the general effects of three typical non-ideal factors on distributed allocation algorithms.

III. Problem Definition

The collaborative visit mission requires finding a conflict-free matching of targets to multi-UAV while minimizing the global mission cost. Each UAV is able to visit a sequence of targets, and each target cost is related to the UAV's target sequences. An allocation is conflict-free if each target is assigned to no more than one UAV [5]. Thus, the task allocation problem investigated in this letter belongs to the SR-ST-ID category. To formulate the problem mathematically, $\mathcal{U}=[u_1,...,u_n]^{\mathrm{T}}$ is defined as the set of n UAVs and $\mathcal{T}=[t_1,...,t_m]^{\mathrm{T}}$ as the set of m stationary targets to be visited. The allocation result is defined as $\boldsymbol{\chi}=[\mathbf{p}_1,...,\mathbf{p}_n]^{\mathrm{T}}$ where $\mathbf{p}_i, i=1,...,n$ is an ordered list of targets assigned to u_i. Generally, the objective of task allocation problems can be divided into min-max form and min-sum form. The min-max form aims to minimize the maximum path cost of each UAV, while the min-sum form aims to minimize the total path cost of all the UAVs. We choose the min-max form here because the targets to be performed in the collaborative visit mission are time critical, so the time cost of each UAV's path is more important [19]. The objective function is expressed as

$$\min_{\boldsymbol{\chi}}\left\{\max_{i\in\mathcal{U}}\sum_{k=1}^{|\mathbf{p}_i|}c_{i,k}\left(\mathbf{p}_i\right)\right\} \tag{1}$$

$$\text{s.t.}\quad \left|\mathbf{p}_i\right|\le C_i \tag{2}$$

$$\bigcup_{i=1}^{n}\mathbf{p}_i=\mathcal{T},\quad \mathbf{p}_i\bigcap\mathbf{p}_j=\varnothing \text{ with } i\neq j \tag{3}$$

where $c_{i,k}(\mathbf{p}_i)$ represents the cost of u_i visiting the k-th target in $\mathbf{p}_i$, and can be calculated by the Dubins [20] length from the $(k-1)$-th target to the k-th one in $\mathbf{p}_i$. The notation $|\cdot|$ denotes the cardinality of the list. Equation (2) represents the capacity constraint of each UAV, ensuring that the number of targets assigned to u_i is less than C_i. Equation (3) guarantees that the allocation result $\boldsymbol{\chi}$ is complete and conflict-free. Assuming the constant velocity of UAVs, the objective of the collaborative visit mission is equivalent to minimizing the mission completion time.

In the collaborative visit mission, multiple UAVs are required to formulate mission plans in advance or in real-time. According to different application stages, task allocation can be divided into pre-allocation and dynamic allocation. In the pre-allocation stage, UAVs have *a priori* knowledge of all the targets' positions and attempt to generate the complete allocation before executing the mission. After the allocation is generated, each UAV visits the corresponding targets along the ordered list according to the allocation results.

In the dynamic allocation, some pop-up targets appear during the collaborative visit mission process. UAVs are required to generate the allocation according to the latest situational awareness within sub-seconds. In this letter, UAVs are set to participate in multiple rounds of reallocation while performing the mission. In each round, a partial allocation is generated within a limited duration. The UAV visits corresponding targets according to its partial allocation and continuously reallocates the remaining targets with other UAVs. Once the aiming target is outbid by another UAV, the UAV adjusts and updates its target list. The dynamic allocation is completed until partial results from multi-UAV can be combined into a conflict-free allocation, i.e.,

$$\mathbf{p}_i = \bigcup_{\lambda=1}^{\lambda^*} \widehat{\mathbf{p}}_i(\lambda), \quad \bigcup_{i=1}^{n} \mathbf{p}_i = \mathcal{T}, \quad \mathbf{p}_i \bigcap \mathbf{p}_j = \varnothing \text{ with } i \neq j \tag{4}$$

where $\widehat{\mathbf{p}}_i(\lambda)$ denotes the partial allocation generated by u_i in λ-th round of reallocation, λ^* represents the round number when the mission is completed.

IV. Algorithms and Communication Models

A. Distributed Task Allocation Algorithms

Among the six comparing algorithms, CBAA, CBBA, HIPC, and PI are market approaches consisting of the task inclusion phase and the consensus phase. In the first phase, the targets are added to the UAV's target list according to certain rules, and the second phase is to reach a consensus for each UAV and remove the outbid targets. The difference between the comparative algorithms is that CBAA is a single-task allocation algorithm, which assigns one target to each UAV. CBBA is the multi-task form of CBAA that allows allocating more than one target to each UAV. The marginal score improvement for u_i greedily including t_k is calculated as

$$w_{i,k}^* = \max_{l \le |\mathbf{p}_i|} \left\{ r_i(\mathbf{p}_i \oplus_l t_k) - r_i(\mathbf{p}_i) \right\} \tag{5}$$

where $r_i(\mathbf{p}_i)$ represents the total reward value for u_i performing the target list $\mathbf{p}_i$, and $\oplus_l$ denotes the operation that inserts the target to the list at the l-th element.

HIPC uses implicit coordination to allocate a subset of the targets to each UAV. Additional information on self-positions is required to accomplish implicit coordination when multi-UAV exchange messages. The nearest neighbor algorithm is utilized to construct the target bundle in this letter.

The marginal significance in PI is devised to measure a target's contribution to the local cost generated by each UAV. The overall cost of the objective can thus be decreased by switching targets amongst different UAVs.

The other two algorithms, DHBA and DGA, are the distributed form of traditional centralized approaches, i.e., the Hungarian algorithm and the Genetic algorithm. As for DHBA, after assigning the target to each UAV using the Hungarian algorithm, each UAV exchanges its cost matrix with its neighbors and updates its cost matrix accordingly. For DGA, each UAV maintains and improves a population of allocation by sharing the current best solutions among the multi-UAVs.

For the pre-allocation problem, once the number of targets is greater than that of UAVs, single-task allocation algorithms (CBAA and DHBA) are unsuitable for dealing with this issue because of $C_i = 1$, while CBBA, HIPC, PI, and DGA are still effective due to the multi-task allocation property. In dynamic allocation, all the six comparative algorithms can be applied to accomplish allocation with multiple rounds of reallocation. The main characteristics of the algorithms are listed in Table I.

TABLE I
DISTRIBUTED TASK ALLOCATION ALGORITHMS COMPARED IN THIS LETTER

Algorithms	Mechanism	Problems Handled	Property	Messages Passed
CBAA	Market-based	Single-task assignment	Deterministic	Winning bid list
CBBA	Market-based	Multi-task assignment	Deterministic	Winning bid list + Winning agent list + Timestamp
HIPC	Market-based	Multi-task assignment	Deterministic	Winning bid list + Winning agent list + Timestamp + UAV Position
PI	Market-based	Multi-task assignment	Deterministic	Winning bid list + Winning agent list + Timestamp
DHBA	Combinatorial optimization	Single-task assignment	Deterministic	Cost matrix
DGA	Swarm heuristic	Multi-task assignment	Stochastic	Current optimal target sequence + UAV Position

B. Communication Models

1) Package Loss Model

A package is defined as a group of messages passed by the UAV in each iteration of the algorithms, according to Table I. Considering the path loss and multipath fading effects of the signal, the package loss model can be formulated as

$$\begin{cases} \text{package is received,} & \gamma_{\text{dB}} \ge P_{\text{bar}} \\ \text{package is lost,} & \gamma_{\text{dB}} < P_{\text{bar}} \end{cases} \tag{6}$$

where P_{bar} denotes the threshold, γ_{dB} represents the SNR model in decibel form, which can be described by a log-normal distribution [16], [21]

$$\gamma_{\text{dB}} = P_T + K_{\text{dB}} - P_{L_0} - 10\alpha \lg(d/d_0) - \mathcal{N}(\mu_{\text{dB}}, \sigma_{\text{dB}}^2) \tag{7}$$

where P_T denotes the transmit power; K_{dB} is the gain based on equipment characteristics; d represents the distance between the transmitter and the receiver; α represents the path loss exponent (equals to 2 in free space, and 4-5 in an environment congested with obstacles and interference [16]); P_{L_0} represents the path loss at the reference distance d_0; $\mathcal{N}(\mu_{\text{dB}}, \sigma_{\text{dB}}^2)$ is the noise term with a mean of μ_{dB} and a variance of σ_{dB}^2. The probability of package loss between u_i and u_k can be calculated by

$$p_{i,k} = \frac{1}{\sigma_{\text{dB}}\sqrt{2\pi}} \int_{-\infty}^{\Theta_{i,k}} \exp\left(-\frac{(x-\mu_{\text{dB}})^2}{2\sigma_{\text{dB}}^2} \right) \mathrm{d}x \tag{8}$$

where $\Theta_{i,k} = P_T + K_{\text{dB}} - P_{L_0} - 10\alpha \lg(d_{i,k}/d_0) - P_{\text{bar}}$, $d_{i,k}$ is the distance between u_i and u_k.

2) Bit Error Model

The received package may still have bit errors, and the BER under BPSK modulation can be expressed as [22]

$$p_e = Q\left(\sqrt{2\gamma}\right) = \frac{1}{2}\operatorname{erfc}\left(\sqrt{\gamma}\right) = \frac{1}{2}\operatorname{erfc}\left(\sqrt{10^{0.1\gamma_{\text{dB}}}}\right) \tag{9}$$

where p_e denotes the BER; $\operatorname{erfc}(\cdot)$ is the complementary error function. Each bit randomly changes between '1' and '0' with the probability p_e around 10^{-6}-10^{-1}. It should be noted that, although p_e is small enough, the bit error probability of the entire package is not negligible due to the large number of bits. Assuming the bit length of the package ϕ is L^{ϕ}, the bit error probability of the entire package is calculated by $1-(1-p_e)^{L^{\phi}}$. When an error occurs at the high byte of the data, the decoding will cause an obvious deviation from the theoretical value. Especially, for the collaborative visit mission, the incorrect decoding of the critical message, such as target costs or identifiers, would degrade the algorithm's optimality and efficiency, or even cause conflicts.

3) Time Delay Model

UAVs are set to send packages to (or receive from) their neighbors within one hop distance. These neighbors then update and transmit the latest information to the next hop. This letter focuses on the time delay of the package transmission in one hop. Referring to the time slot allocation method combined with election and reservation in the IEEE 802.16 protocol [23], the package transmission delay τ_{hop} under the TDMA distributed resource scheduling mechanism can be expressed as

$$\tau_{\text{hop}} = (\hat{h} + \frac{1}{p_{\text{ele}}})T_{\text{slot}}, \quad \frac{1}{p_{\text{ele}}} = (N_{\text{2-hop}} - 1) \times \frac{\hat{v} + \frac{1}{p_{\text{ele}}}}{\hat{h} + \frac{1}{p_{\text{ele}}}} + 1 \quad (10)$$

where T_{slot} represents the time slot, p_{ele} denotes the election probability of the current UAV, $\hat{v}$ and $\hat{h}$ are election backoff parameters, $N_{\text{2-hop}}$ denotes the number of neighbor UAVs within two hops. From (10), one finds that the time delay of the package transmission in one hop increases as the number of surrounding neighbors grows. The one-hop distance can be calculated as

$$d_{\text{hop}} = d_0 \cdot 10^{\frac{P_T + K_{\text{dB}} - P_{L_0} - P_{\text{bar}} - \mu_{\text{dB}} - \sigma_{\text{dB}}}{10\alpha}} \quad (11)$$

where d_{hop} denotes the distance that the received signal power equals to P_{bar} considering the noise. When the distance between UAVs exceeds d_{hop}, the package loss probability is approximately 1.

V. ALGORITHM CUSTOMIZATION

To satisfy the requirements of different allocation stages (pre-allocation or dynamic allocation), the referred algorithms are customized in this section. The customization involves communication modes, bid calculation, convergence conditions, and evaluation metrics.

A. Communication Modes

Generally, the communication mode of task allocation algorithms can be classified into synchronous mode and asynchronous mode. In the pre-allocation, task allocation algorithms can operate synchronously during the allocation process, i.e., each UAV exchanges bids, significance, or best solutions with other UAVs within the same iteration. The synchronous consensus phase of HIPC and PI is inherited from CBBA [5]. To ensure enough messages from other UAVs are collected and avoid the endless waiting caused by harsh environments, the mechanism of *maximum bid wait time* [24] is applied in each iteration.

In dynamic allocation, however, the adoption of synchronous communication may not be feasible due to strict constraints on time-consuming and bandwidth. Asynchronous communication is applied for the allocation algorithms in which the message-transmitting thread runs separately from the allocation thread. The UAV stores and selects the latest messages from each neighbor to update its target list accordingly. The consensus protocol applied to the comparative algorithms (CBBA, HIPC, and PI) should be revised in the presence of asynchronous updates (refer to ACBBA in [6]).

B. Bid Calculation

Considering the min-max objective function, the bid calculation for target t_k of auction-based allocation algorithms is designed as the distance that u_i travels along the path $\mathbf{p}_i$ including (or excluding) t_k. As for PI, in particular, since it directly optimizes the objective function, two additional lists ($\boldsymbol{\varpi}_i$ and $\boldsymbol{\phi}_i$) need to be updated and added to the message transmission between UAVs. The k-th element in $\boldsymbol{\varpi}_i$ represents the path cost of the UAV winning target t_k, and in $\boldsymbol{\phi}_i$ represents the remained path cost when the UAV loses t_k. The bid (significance) of t_k included in the target list $\mathbf{p}_i$ with regard to u_i should be modified as

$$w_{i,k}(\mathbf{p}_i \ominus t_k) = \max\left\{\boldsymbol{\varpi}_i(k), \sum_{r=1}^{|\mathbf{p}_i|-1} c_{i,r}(\mathbf{p}_i \ominus t_k)\right\} - \max\left\{\boldsymbol{\phi}_i(k), \sum_{r=1}^{|\mathbf{p}_i|} c_{i,r}(\mathbf{p}_i)\right\} \quad (12)$$

where $\mathbf{p}_i \ominus t_k$ represents removing t_k from $\mathbf{p}_i$.

C. Convergence Conditions

The convergence condition for the pre-allocation is designed as

$$\left\{\boldsymbol{\chi}^{(\varsigma)} = \boldsymbol{\chi}^{(\varsigma-\tau)}, \tau = 1,2,...,\mathcal{D}\right\} \text{ or } \left\{\varsigma \geq \varsigma_{\max}\right\} \quad (13)$$

where $\mathcal{D}$ denotes the network diameter [5], $\boldsymbol{\chi}^{(\varsigma)}$ represents the allocation result in the ς-th iteration, and $\varsigma_{\max}$ denotes the maximum iteration count. On the other hand, for the dynamic allocation, the convergence condition is expressed as

$$\bigcup_{i=1}^{n}\bigcup^{\lambda} \hat{\mathbf{p}}_i(\lambda) = \bigcup_{i=1}^{n}\bigcup^{\lambda-\tau} \hat{\mathbf{p}}_i(\lambda), \quad \tau = 1,2,...,\mathcal{D} \quad (14)$$

D. Evaluation Metrics

Furthermore, to thoroughly evaluate algorithm performance, four metrics (optimality, communication overhead, efficiency, and reliability) are chosen, as shown in (15). Among them, the optimality metric $\mathcal{M}_{\text{opt}}$ is calculated by the path length of UAVs after finishing the mission; the communication overhead metric $\mathcal{M}_{\text{com}}$ is measured by the transmitted package number and the bit length; the efficiency metric $\mathcal{M}_{\text{eff}}$ is calculated by the algorithm runtime; the reliability metric $\mathcal{M}_{\text{cfl}}$ is measured by the conflict allocation rate.

$$\begin{cases} \mathcal{M}_{\text{opt}} = \max_{i\in\mathcal{U}} \sum_{k=1}^{|\mathbf{p}_i|} c_{i,k}(\mathbf{p}_i) \\ \mathcal{M}_{\text{com}} = \max_{i\in\mathcal{U}} \left(N_i^{\text{pkg}} L^{\phi}\right) \\ \mathcal{M}_{\text{eff}} = \max_{i\in\mathcal{U}} \Delta_i \\ \mathcal{M}_{\text{cfl}} = \Upsilon(\boldsymbol{\chi})/\Xi \end{cases} \quad (15)$$

where N_i^{pkg} denotes the total number of transmitted packages of u_i, L^{ϕ} denotes the bit length of the transmitted package ϕ, Δ_i denotes the algorithm runtime on u_i, $\Upsilon(\boldsymbol{\chi})$ represents the number of conflict allocations in Monte Carlo simulations, Ξ represents the total simulation times.

VI. Experimental Results and Discussion

The simulation framework consisting of UAV and Communication modules is built using the Robot Operating System (ROS) Noetic. The UAV module carries out the task allocation algorithms in a distributed way. The Communication module simulates the message exchanges and decides whether the passing messages are received or lost, correct or error, and punctual or delayed. The pre-allocation scenario with synchronous communication and the dynamic allocation scenario with asynchronous communication are performed. All simulations are implemented on an AMD Ryzen 7 3700X, 8-core, 3.59 GHz CPU with 16 GB RAM.

A. Experimental Setup

The locations of the UAVs and targets are randomly generated in the 2D mission area ([0, 3000]m, [0, 3000]m). The velocity of homogeneous UAVs is set to be 80m/s, and the minimum turning radius is 200m. When the UAV reaches the threshold distance of 0.5m of the target's location, the target is considered to be *visited*. The communication model parameters are chosen as follows: $P_{L_0} = 80\text{dB}$, d_0=100m, $P_T = 30\text{dBm}$ [2], $K_{\text{dB}} = 30\text{dB}$, $\mu_{\text{dB}} = -70\text{dBm}$, $\sigma_{\text{dB}} = 7\text{dB}$, $P_{\text{bar}} = 5\text{dB}$, $\hat{v}=2$, $\hat{h}=32$ and T_{slot}=7.8ms [25]. The path loss exponent α ranges from 2.0 to 4.0 representing different communication qualities. For convenience, we define $\alpha=2.0$ and 3.0 as mild communication conditions and $\alpha=4.0$ as harsh communication conditions. The one-hop time delay is around 0.2-0.3s.

In both pre-allocation and dynamic allocation scenarios, shown in Table II, two different allocation scales are considered, in which non-ideal communication factors are gradually added. Each case runs 100 Monte Carlo simulations. In the pre-allocation scenario, the capacity C_i of each UAV is chosen as the number of targets. The maximum iteration count $\varsigma_{\max}$ for these algorithms is set to 100. The *maximum bid wait time* in one iteration is set to 0.5s. For the dynamic scenario, $\varsigma_{\max}$ is set up to 2 for all the six algorithms. The capacity C_i of the multi-task allocation algorithms is set to 5. The dynamic reallocation period is set to 0.1s.

TABLE II
Different Issues in Each Scenario

Scales & Cases	Issues
Scale 1	4 UAVs and 10 targets (n=4 m=10)
Scale 2	7 UAVs and 18 targets (n=7 m=18)
Case 1	Only package loss factors are considered
Case 2	Package loss and bit error factors are both considered
Case 3	Three non-ideal factors are considered

B. Pre-allocation Scenario

In the pre-allocation scenario, four multi-task allocation algorithms (i.e., CBBA, HIPC, PI, DGA) are compared in two problem scales with three different cases. The maximum path length of the conflict-free allocations is illustrated in Fig. 1. Overall, the optimality of each algorithm does not change significantly in different cases with varying α. Among them, DGA has the best optimality, thus yielding the shortest boxplot. Besides, the optimality of PI is superior to CBBA and HIPC. It is because that CBBA optimizes the local objective under the diminishing marginal gain (DMG) assumption, and HIPC uses the greedy nearest neighbor algorithm to generate allocations, which decreases global optimality. On the contrary, DGA explores the global space better via probabilistic inheritance and mutation, and PI directly optimizes the global min-max optimization objective.

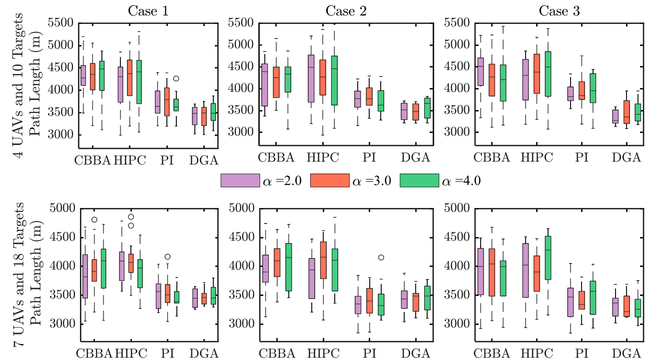

Fig. 1. Optimality comparison of the multi-task allocation algorithms. The lower and shorter box means the better optimality.

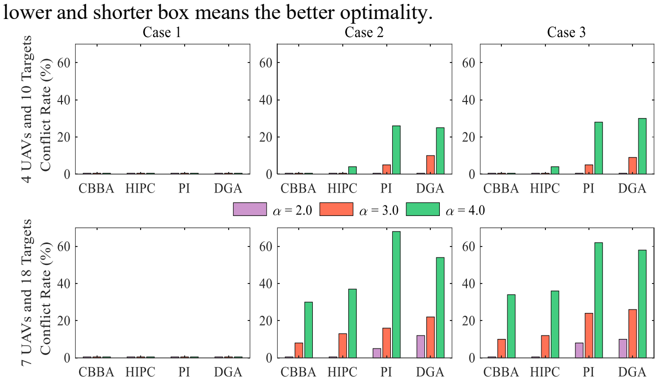

Fig. 2. Conflict rate of each algorithm under non-ideal communication.

To be mentioned, the data used for optimality statistics are collected from conflict-free allocations. The ratio of conflicted allocations in comparison results is shown in Fig. 2. Without considering bit error in *Case 1*, the algorithms generate conflict-free results in both scale problems. However, after bit error factors are additionally considered in *Case 2* and *Case 3*, the conflicted allocation results exist. As α increases, the conflict rate of each algorithm rises simultaneously. Specifically, DGA has a probability of nearly 10% to get a conflict allocation with α=3.0 when solving *Scale 1* problem, and it reaches a probability of 30% if $\alpha = 4.0$. As for *Scale 2* issue, DGA suffers from more than 50% probability of conflicted allocation with α=4.0, which means that DGA cannot work reliably in such a non-ideal communication environment. The main reason is that the UAV running DGA evolves its target sequence list through the best allocation sequence from neighbor UAVs, and the message carrying the target sequence is vulnerable to external interference because of the changeable target identifier. PI has a similar conflict performance as DGA. With mild communication conditions (α=2.0 and 3.0), the probability of conflict in PI is below 25%. When α=4.0, the conflict rate reaches nearly 30% and 65% for *Scale 1* and *Scale 2*, respectively. The reason is that PI needs to transmit two additional messages ($\boldsymbol{\varpi}_i$ and $\boldsymbol{\phi}_i$) for the min-max objective, as mentioned earlier, which increases the probability of error and conflict of critical information with harsh communication conditions (α=4.0). HIPC has a relatively lower conflict rate, with nearly 5% and 40% for *Scale 1* and *Scale 2*, respectively. Among the competitors, CBBA has the best reliability. For *Scale 1*, the conflict probability of CBBA is approximately 0. And for *Scale 2*, the conflict probability of CBBA is also the lowest compared with the competitors. Therefore, the reliability order of the algorithms considering bit error factors is summarized as CBBA > HIPC > DGA $\approx$ PI.

The communication overhead and efficiency metrics are illustrated in Table III (*Case 2* presents similarly to *Case 1 and Case 3*, thus omitted). According to metrics' values, all the algorithms have increased the overhead and runtime with the degradation of communication, since more transmitted messages are required and more iterations are needed to compensate for the effects of message loss, error, and delay. Generally, the values in *Case 1* are lower than those in *Case 3* under the same conditions. As the problem scale and α increase, the value gap between *Case 1* and *Case 3* becomes larger. For the communication metric, CBBA has the lowest overhead compared with the competitors because the bit length of CBBA packages is shorter, according to Table I. HIPC also performs well at the communication metric since it requires sharing current locations among the UAVs to solve a global allocation on each local platform, thus yielding a better consensus than other algorithms due to fewer iterations.

Algorithm performance in the efficiency metric is similar to the communication one. For *Scale 1*, CBBA and HIPC both converge in a few seconds with mild communication conditions, and HIPC performs the best in efficiency. Notice that the impact of time delay and bit error factors on HIPC runtime increment is smaller than that of CBBA, confirming the fewer iterations of HIPC with mild communication conditions. On the contrary, DGA requires tens of seconds to reach a feasible allocation. As the communication quality degrades, the mean value and standard deviation of CBBA runtime are lower than HIPC when $\alpha=4.0$, indicating a more stable performance of CBBA in harsh communication conditions.

Although DGA and PI can obtain better optimality solutions, they require more communication and computing resources. Moreover, a longer runtime may lead to more risks of conflicts. To sum up, CBBA generally outperforms the others during the pre-allocation stage with better reliability, higher efficiency, and less communication overhead.

Additionally, to further discuss the algorithm efficiency, the effects of non-ideal communication factors on the algorithm runtime are analyzed. The theoretical runtime of the distributed task allocation algorithm can be expressed as

$$\mathcal{R} = \mathcal{R}_0\left(\varsigma_{\max}\right) + \sum_{\varsigma=1}^{\varsigma^*} \max_{i\in\mathcal{U}} \tau^i_{\varsigma,\mathrm{hop}} \tag{16}$$

where $\mathcal{R}_0$ denotes the runtime without considering time delay factors, $\sum_{\varsigma=1}^{\varsigma^*}\max_{i\in\mathcal{U}}\tau^i_{\varsigma,\mathrm{hop}}$ is the theoretical time delay term, and $\tau^i_{\varsigma,\mathrm{hop}}$ denotes the one hop delay of u_i at iteration ς. The total iteration count ς^* and one hop delay $\tau^i_{\varsigma,\mathrm{hop}}$ are two critical factors affecting the algorithm runtime.

Clearly, $\tau^i_{\varsigma,\mathrm{hop}}$ within the *maximum bid wait time* leads to limited effects on ς^* since the delayed messages will finally be received in each iteration. Thus, the algorithm runtime is approximately linearly related to the maximum one-hop delay in the system with a constant ς^*. In this letter, the one-hop delay is at the sub-second level, and the incremental runtime affected by time delay is thus limited. However, when α rises, package loss and bit error factors may cause rapid growth of ς^*, leading to $\mathcal{R}_0$ and $\sum_{\varsigma=1}^{\varsigma^*}\max_{i\in\mathcal{U}}\tau^i_{\varsigma,\mathrm{hop}}$ terms both increasing, which results in a long convergence time. Thus, we conclude that, under harsh communication conditions, package loss and bit error factors significantly affect the algorithm runtime compared with time delay factors. Fig. 3 shows the algorithm's runtime with or without considering time delay (*Case 3* and *Case 2*). The stack effect of time delay factors on runtime is close to the theoretical analysis. When α varies from 3.0 to 4.0, the proportion of the incremental runtime of CBBA, HIPC, and PI caused by loss and error factors has reached 75.5%, 78.0%, and 81.7%, respectively. In contrast, the proportion affected by time delay is only 24.5%, 22.0%, and 18.3%, respectively.

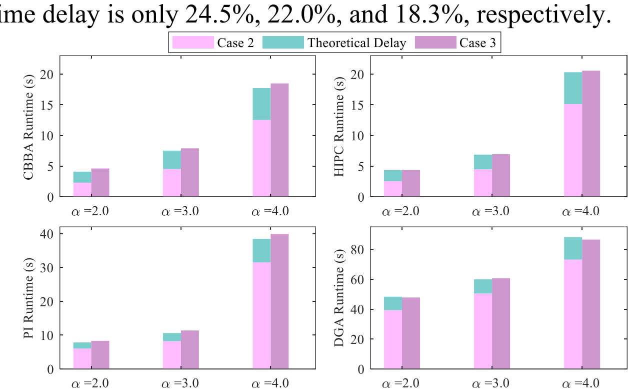

Fig. 3. Runtime comparison among different cases considering the theoretical time delay in Scale 1 problem.

TABLE III
COMPARISON OF COMMUNICATION OVERHEAD AND EFFICIENCY METRICS IN CASE 1 AND CASE 3

	Algorithms	$\alpha=2.0$				$\alpha=3.0$				$\alpha=4.0$			
		$\mathcal{M}_{\mathrm{com}}$ ($\times10^4$ bit)		$\mathcal{M}_{\mathrm{eff}}$ (s)		$\mathcal{M}_{\mathrm{com}}$ ($\times10^4$ bit)		$\mathcal{M}_{\mathrm{eff}}$ (s)		$\mathcal{M}_{\mathrm{com}}$ ($\times10^4$ bit)		$\mathcal{M}_{\mathrm{eff}}$ (s)	
		Case 1	*Case 3*	*Case 1*	*Case 3*	*Case 1*	*Case 3*	*Case 1*	*Case 3*	*Case 1*	*Case 3*	*Case 1*	*Case 3*
Scale 1 ($n=4$ $m=10$)	CBBA	**0.96**	**1.08**	2.28	4.62	**1.10**	**1.58**	3.51	7.90	**2.15**	**2.79**	**10.54**	**18.48**
	HIPC	1.08	1.22	**2.12**	**4.39**	1.28	1.63	**3.11**	**6.94**	3.52	3.62	14.90	20.56
	PI	2.09	2.19	5.88	8.32	2.71	3.03	7.60	11.37	7.34	8.06	31.44	39.93
	DGA	3.39	3.63	37.17	47.79	3.98	4.39	45.73	60.68	5.65	5.71	71.78	86.51
Scale 2 ($n=7$ $m=18$)	CBBA	**3.56**	**3.70**	8.32	13.02	**4.59**	**5.00**	12.97	19.30	**8.04**	**9.36**	**30.58**	**42.51**
	HIPC	4.33	4.61	**7.65**	**12.07**	5.26	6.12	**11.93**	**18.52**	10.32	13.52	30.65	49.33
	PI	7.28	7.62	13.73	20.74	8.42	11.25	23.07	32.78	13.36	20.16	41.57	69.28
	DGA	8.85	9.58	221.38	241.33	11.06	11.98	219.56	240.71	13.03	12.30	285.39	251.20

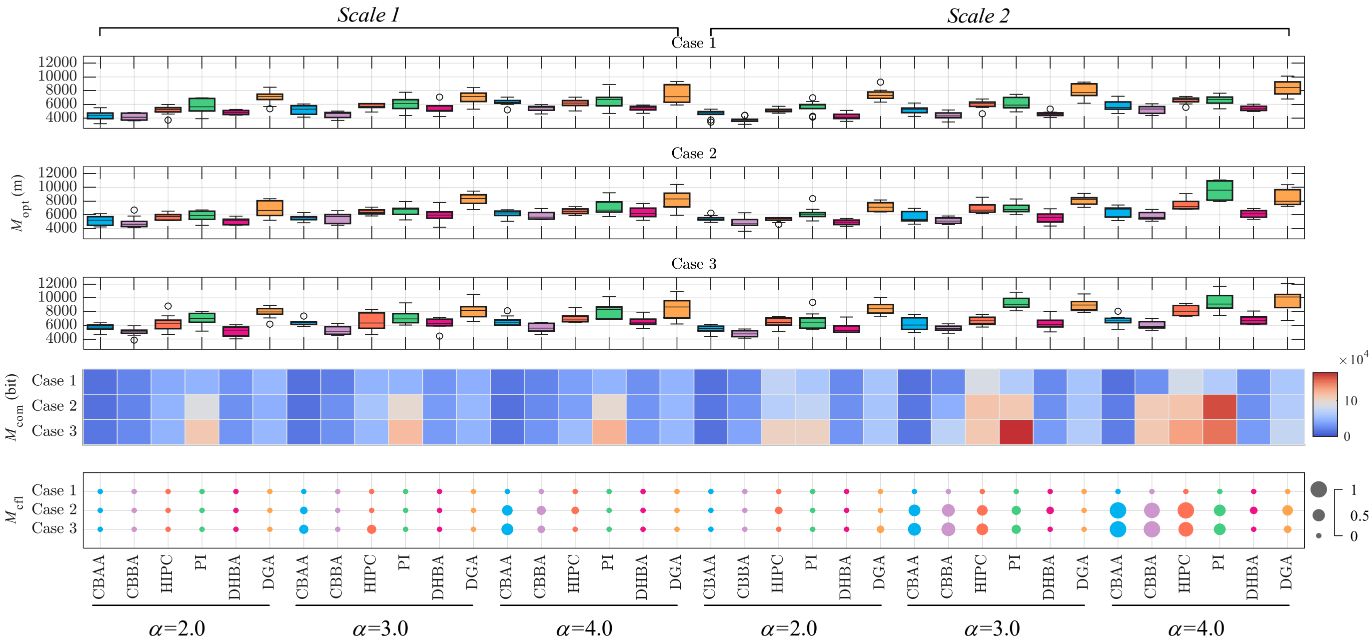

Fig. 4. Comparison results of the task allocation algorithms in the dynamic scenario. The optimality, communication overhead, and reliability metrics are selected to evaluate allocation performance (the efficiency metric is omitted because the algorithms generate the allocation within 0.1s in each round). The first three subgraphs correspond to optimality, the fourth corresponds to communication overhead, and the last corresponds to reliability.

C. Dynamic Allocation Scenario

In the dynamic scenario, the targets are classified into pre-known and pop-up. The number of pop-up targets is randomly selected in the range of $[0,5]$ for both scale problems during the mission execution. The comparison results of the algorithms in three cases are shown in Fig. 4. The left part of Fig. 4 is the results of *Scale 1*, and the right part contains the results of *Scale 2*.

For the optimality metric, the minimum path length of each UAV increases as α and non-ideal communication factors gradually increased, which is quite different from the pre-allocation scenario. The reason is that the UAV often loses and updates the aiming targets due to the abnormal messages caused by package loss and bit error factors. The outbidding and reallocating make the UAV turn back and forth between targets, and greatly degrades allocation optimality. Obviously, the optimality in the dynamic allocation is worse than that in the pre-allocation due to the lacking of global coordination in each round. Among the comparative algorithms, CBBA performs the best with the shortest path lengths. The optimality of CBAA and DHBA is slightly weaker than CBBA. DGA sacrifices optimality to solve the dynamic problem in real-time.

For the reliability metric, allocation conflicts in the dynamic scenario mainly manifest as *unvisited* targets. As α grows, the number of *unvisited* targets gradually increases. Without considering bit error factors in *Case 1*, all the six algorithms generate conflict-free allocations in both scale problems, which is consistent with the pre-allocation scenario. After the time delay is considered in *Case 3*, the effects of time delay on conflicts for each algorithm are quite limited compared with that of bit error in *Case 2*. Summarily, DHBA and DGA generate fewer conflicts in harsh communication conditions.

For the communication metric, the single-task allocation algorithms (CBAA and DHBA) perform better than the multi-task allocation algorithms under harsh communication conditions since the bit length of the transmitted package is shorter. For the PI algorithm, the additional messages lead to a high communication overhead. To sum up, CBBA and DHBA generally outperform the other competitors in terms of optimality, communication overhead, and reliability in the dynamic scenario. Specifically, CBBA has the best optimality, and DHBA generates less communication overhead and fewer conflicts in harsh environments.

To further analyze the effects of different communication modes on the allocation algorithms, the efficiency and optimality performance of the comparative algorithms under synchronous/asynchronous communication are discussed. For the efficiency part, note that the one-hop time delay is longer than the dynamic reallocation period (0.1s). From (16), one finds that the incremental runtime affected by time delay factors in each iteration is much larger than the reallocation period in synchronous mode. Thus, all the comparative algorithms can hardly solve the dynamic allocation within 0.1s. In contrast, the algorithms can still generate allocations in time under the asynchronous communication in *Case 3,* which indicates that asynchronous mode can eliminate the delay's impact. It is because each UAV starts the allocation process and updates the local states independently without global synchronization, the inter-UAV message transmission delay is thus avoided. However, due to the lack of global consensus assurance, asynchronous communication may reduce the optimality of distributed allocation algorithms.

To compare the optimality of different communication modes, synchronous communication is temporarily applied to the dynamic scenario with the algorithm runtime latency omitted caused by the global consensus. Monte Carlo simulations are conducted to compare the maximum path length generated by algorithms with different communication modes on *Case 3* in *Scale 1*. As $\alpha = 2.0$, the path length of the algorithms in the asynchronous mode is increased at most by 5.3% compared with that in the synchronous mode. When α rises to 4.0, the maximum proportion decreases to 3.2%, indicating that asynchronous communication reduces the optimality of distributed allocation algorithms in the dynamic scenario. As the communication quality degrades, the inferiority of the asynchronous mode is gradually diminished because the large number of lost packages and error bits in harsh environments weaken the global consensus of synchronous communication. In conclusion, asynchronous communication alleviates the effects of time delay during the distributed allocation and is more suitable for the dynamic scenario compared with synchronous communication.

VII. Conclusion

This letter compares and analyzes the effects of three non-ideal communication factors (loss, error, and delay) on the performance of several typical distributed task allocation algorithms through simulation. Under the pre-allocation scenario, CBBA performs best at reliability, communication overhead, and efficiency. For the dynamic scenario, CBBA has advantages in optimality, while DHBA exhibits less overhead and fewer conflicts in harsh environments.

We concluded that bit error factors would cause conflicts violating the allocation constraints and degrade the task allocation algorithm performance. The effect of time delay on the algorithm runtime is approximately the accumulation of single hop delay in multiple iterations under synchronous communication. The algorithm efficiency will not be significantly affected if the time delay is limited at the sub-second level. However, the algorithm iteration count increases sharply due to package loss and bit error factors in harsh environments, which actually affects the efficiency. We suggest that reducing the impact of loss and error factors is an effective way to enhance the algorithm's efficiency in non-ideal communication environments. Asynchronous communication can eliminate time delay but sacrifices the optimality of the algorithm, which is more suitable for real-time dynamic environments.

In future research, we will focus on designing a suitable allocation mechanism to reduce the impact of bit errors and attempt to improve the optimality when applying asynchronous communication in a dynamic scenario.

Acknowledgment

The multiagent simulation framework used in this letter is based on Dr. Sharan Nayak's previous work.

References

[1] J. Xiong, J. Li, J. Li, S. Kang, C. Liu, and C. Yang, "Probability-tuned market-based allocations for uav swarms under unreliable observations," *IEEE T. Cybern.*, 2022. [Online]. Available: https://doi.org/10.1109/TCYB.2022.3162907

[2] S. Nayak, S. Yeotikar, E. Carrillo, E. Rudnick-Cohen, M. Jaffar, R. Patel, S. Azarm, J. W. Herrmann, H. Xu, and M. Otte, "Experimental comparison of decentralized task allocation algorithms under imperfect communication," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 572–579, Apr. 2020.

[3] S. Ismail, and S. Liang, "Decentralized hungarian-based approach for fast and scalable task allocation," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, Miami, 2017, pp. 23–28.

[4] R. Patel, E. Rudnick-Cohen, S. Azarm, M. Otte, and J. W. Herrmann, "Decentralized task allocation in multi-agent systems using a decentralized genetic algorithm," in *Proc. IEEE Int. Conf. Rob. Autom.*, Paris, 2020, pp. 3770–3776.

[5] H. L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 912–926, Sept. 2009.

[6] L. B. Johnson, S. S. Ponda, H. L. Choi, and J. P. How, "Improving the efficiency of a decentralized tasking algorithm for uav teams with asynchronous communications," in *Proc. AIAA Guid., Navigation, Control Conf.*, Toronto, 2010, pp. 8421.

[7] L. Johnson, H. L. Choi, and J. P. How, "The hybrid information and plan consensus algorithm with imperfect situational awareness," *Distributed Autonomous Robotic Systems*, in *Proc. Distrib. Auton. Robot. Syst.*, Tokyo, 2016, pp. 221–233.

[8] W. Zhao, Q. Meng, and P. Chung, "A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario," *IEEE T. Cybern.*, vol. 46, no. 4, pp. 902–915, April 2015.

[9] D. D. Castañon, "Parallel synchronous and asynchronous," *Parallel Comput.*, vol. 17, no. 6–7, pp. 707–732, Sept. 1991.

[10] 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, Sept. 2004.

[11] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," in *Proc. IEEE Control Syst.*, Mexico, 2008, pp. 1212–1217.

[12] G. Gao, Y. Mei, Y. H. Jia, W. N. Browne, and B. Xin, "Adaptive coordination ant colony optimization for multipoint dynamic aggregation," *IEEE T. Cybern.*, vol. 52, no. 8, pp. 7362–7376, Aug. 2022.

[13] Z. Han, A. L. Swindlehurst, and K. R. Liu, "Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3533–3546, Feb. 2009.

[14] E. Carrillo, S. Yeotikar, S. Nayak, M. K. M. Jaffar, S. Azarm, J. W. Herrmann, M. Otte, and H. Xu, "Communication-aware multi-agent metareasoning for decentralized task allocation," *IEEE Access*, vol. 9, pp. 98712–98730, Jul. 2021.

[15] A. Bapat, B. R. Bora, J. W. Herrmann, S. Azarm, H. Xu and M. Otte, "Distributed task allocation algorithms for multi-agent systems with very low communication," *IEEE Access*, vol. 10, pp. 124083-124102, Nov. 2022.

[16] A. Kopeikin, S. S. Ponda, L. B. Johnson, and J. P. How, "Multi-uav network control through dynamic task allocation: ensuring data-rate and bit-error-rate support," in *Proc. IEEE Globecom Workshops*, Anaheim, 2012, pp. 1579–1584.

[17] S. Raja, G. Habibi, and J. P. How, "Communication-aware consensus-based decentralized task allocation in communication constrained environments," *IEEE Access*, vol. 10, pp. 19753–19767, Dec. 2022.

[18] J. Li, R. Chen, "A distributed task scheduling method based on conflict prediction for ad hoc uav swarms," *Drones-Basel*, vol. 6, no. 11, pp. 356, Nov. 2022.

[19] A. Prasad, H. L. Choi, S. Sundaram, "Min-max tours and paths for task allocation to heterogeneous agents," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 3, pp. 1511–1522, Sept. 2020.

[20] Z. Chen, "On dubins paths to a circle," *Automatica*, vol. 117, pp. 108996, Jul. 2020.

[21] Y. Yan, and Y. Mostofi, "Robotic router formation in realistic communication environments," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 810–827, Aug. 2012.

[22] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[23] M. Cao, W. Ma, Q. Zhang, and X. Wang, "Analysis of IEEE 802.16 mesh mode scheduler performance," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1455–1464, Apr. 2007.

[24] D. H. Lee, S. A. Zaheer, and J. H. Kim, "A resource-oriented, decentralized auction algorithm for multirobot task allocation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1469–1481, Oct. 2015.

[25] C. I. Cruz, "Netwars based study of a joint stars link-16 network," M.S. thesis, Dept. Electr. Comput. Eng., Air Force Inst. of Tech., Wright-Patterson Air Force Base, Ohio, 2004.