ITRA: Incremental Task Replanning Algorithm for multi-UAV Based on Centralized-Distributed Negotiation

Peng Tian[†], Xiangyu Shan[†], Xinpeng Lu, Xueqing Li, and Junwu Zhu[(✉)]

College of Information Engineering, Yangzhou University, Yangzhou, China
{mz120230979,231305205,211301216,dx120200080}@stu.yzu.edu.cn
jwzhu@yzu.edu.cn

Abstract. Unmanned Aerial Vehicles (UAVs) are an emerging novel type of equipment that accomplish predetermined goals through task preplanning by the base station (BS). However, with the increasing application of UAVs, they face challenges such as changing task environments, high communication pressure on BS, and high time complexity of global planning algorithms during task execution. To address these issues, collaborative task replanning for multi-UAV is required. Therefore, this paper proposes an Incremental Task Replanning Algorithm (ITRA) for multi-UAV based on centralized-distributed negotiation to solve the problem of dynamic task planning. By establishing periodic communication of UAVs in a centralized-distributed mode, the real-time sharing of UAV information and self-organizing management are achieved, enabling multi-UAV to quickly organize autonomously to adapt to environmental changes. Based on this, we designed a centralized-distributed negotiation incremental replanning model. The model seeks feasible solutions with a minimal scheduling scale by gradually increasing the number of UAVs involved in replanning, avoiding extensive scheduling. Experimental comparisons and validations demonstrate that this method effectively handles unexpected situations in dynamic environments and achieves higher efficiency compared to global replanning under the same conditions.

Keywords: Multi-UAV · Centralized-Distributed Architecture · Incremental Task Replanning · Periodic Communication

1 Introduction

In recent years, UAVs have proven valuable in areas like search and rescue, smart agriculture [1,2], environmental monitoring [3], and military operations [4]. Multi-UAV collaboration offers greater reliability, range, and scalability compared to single UAV, especially for complex and dynamic tasks. Effective planning and replanning are key to successful execution in such environments.

Task preplanning, typically handled by the base station (BS), determines task sequences for each UAV based on global information. However, two major

[†] First Author and Second Author contribute equally to this work.

challenges arise: the lack of efficient communication and coordination between UAVs and the difficulty in addressing uncertainties in dynamic environments.

Task replanning addresses these issues, allowing UAVs to adapt to changes, optimize tasks, and ensure system robustness. Current approaches, however, often rely on centralized systems that overwhelm the BS with data, leading to communication bottlenecks and delays. Global task rearrangement algorithms also suffer from high computational demands, limiting their real-time effectiveness in large-scale UAV swarms.

Several communication architectures have been proposed to tackle these issues, including centralized, distributed, and hybrid models. **Centralized architectures**, such as that used by Wu et al. [5], rely on a central node for control but struggle with scalability. **Distributed architectures**, **l**ike the one in the literature [6], allow UAVs to make autonomous decisions based on local information, making them ideal for dynamic environments. **Hybrid architectures**, which blend both approaches, improve coordination in large UAV clusters [7], while multi-UAV clustering, as proposed by literature [8], boosts transmission efficiency by optimizing local information relay.

Replanning mechanisms also vary. Like resource-constrained planning [9], focus on self-sufficient UAVs and resource sharing, while others, such as the distributed architecture in the literature [10], adjust to environmental changes dynamically. The dynamic ant colony labor division model proposed in the literature [11] further enhances UAV autonomy through effective information sharing.

To address the aforementioned issues, we propose a new UAV task replanning scheme. This scheme achieves information sharing among UAVs through periodic communication and uses a centralized-distributed negotiation algorithm to reduce the number of UAVs involved in replanning, thereby achieving higher efficiency. The main contributions of this paper are as follows:

- A dynamic temporary UAV cluster is designed in a centralized-distributed framework, where the cluster head plans tasks and the members execute them based on their own conditions.
- Establishing a periodic communication mechanism ensures information sharing among UAVs, improving coordination and task execution reliability.
- The incremental negotiation mechanism aims to find a feasible solution within the current UAV configuration in the dynamic environment.
- The proposed task replanning model is functionally complete and effectively handles dynamic environments, resulting in higher utility as verified by experiments conducted in two types of dynamic environments.

The paper includes a formal model expression in Section 2, the proposed algorithm in Section 3, experimental validation in Section 4, and conclusions in Section 5.

2 System Model

In this section, we provide a detailed introduction to the network architecture design and problem formulation for the multi-UAV task replanning.

2.1 Centralized-Distributed Architecture

We present a centralized-distributed architecture for multi-UAV task replanning, which involves BS and UAVs. The coordination among the UAVs is achieved through satellite positioning and wireless communication systems.

In this architecture, multi-UAV task planning consists of two stages: preplanning and replanning. During the **preplanning stage**, UAVs follow predefined routes without $BS's$ control to ensure decentralization. In the **replanning stage**, UAVs form small clusters with cluster heads managing task replanning.

Responsibilities are divided as follows: the BS monitors flight without direct control over UAVs, cluster heads manage tasks for cluster member UAVs, including information exchange and task replanning, and cluster members communicate with the cluster head for information transfer and replanning outcomes.

2.2 Problem Description

In the centralized-distributed architecture of multi-UAV systems, there are a BS, N_T tasks $\mathcal{T} = \{T_1, \ldots, T_i, \ldots, T_{N_T}\}$, and N_U UAVs $\mathcal{U} = \{U_1, \ldots, U_j, \ldots, U_{N_U}\}$. Additionally, it is specified that $N_U < N_T$. At time t, T_i is denoted by $T_i^t = \langle l_i, v_i, e_i \rangle$, where $l_i \in \mathbb{R}^2$ denotes the task location, v_i denotes the task value, e_i denotes the energy consumption required to complete the task. Similarly, U_j is denoted by $U_j^t = \langle l_j^t, P_j^t, e_j^t \rangle$ where $l_j^t \in \mathbb{R}^2$ denotes the location of U_j at time t, $P_j^t = \{p_{j,1}, \ldots, p_{j,l}, \ldots, p_{j,|P_j^t|}\}$ denotes the sequence of tasks planned to U_j, where $p_{j,l} \in P_j^t$ and $P_j^t \subseteq \mathcal{T}$. e_j^t denotes the remaining energy of U_j. Specifically, at $t = 0$, l_j^0 denotes the starting point of U_j, P_j^0 is the preplanned task sequence for U_j, and e_j^0 denotes the initial energy of U_j, which is its maximum capacity. It is worth noting that our work focuses on the process of UAVs task replanning rather than flight trajectories. Therefore, we assume that UAVs fly at a constant speed between each task point.

In the task preplanning stage, P_j^t for each U_j satisfies the following constraints: (1) $\cup_{j=1}^{N_U} P_j^t = \mathcal{T}$,which denotes that all tasks must be completed. (2) $\cap_{j=1}^{N_U} P_j^t = \emptyset$, denotes no task is completed by multi-UAV simultaneously. (3) $P_j^t \neq \emptyset$, denotes each UAV has at least one task planned. During the task replanning stage, considerations include dynamic environments such as new tasks and damaged UAVs. Suppose at time t', U_j discovers a set of new tasks $\mathcal{T}'$ or a set of damaged UAVs $\mathcal{U}'$. The updated set of tasks to be completed in the system becomes $\mathcal{T} = \cup_{j=1}^{N_U} P_j^{t'} \cup \mathcal{T}'$, and the UAVs set is updated to $\mathcal{U} \backslash \mathcal{U}'$. The updated task sequence for U_j after the update is denoted as $P_j^{t'}, U_j \in \mathcal{U} \backslash \mathcal{U}'$. Similarly, the updated $P_j^{t'}$ must satisfy the forementioned three constraints. ΔP_j denotes completed tasks by U_j at time t'. The lengths of flights flown by U_j are given:

$$D_j^1 = d(l_j^0, p_{j,1}) + d(p_{j,|\Delta P_j|}, l_j^{t'}) + \sum_{l=1}^{|\Delta P_j|-1} d(p_{j,l}, p_{j,l+1}), p_{j,l} \in P_j^0, p_{j,l} \in \mathcal{T} \quad (1)$$

$$D_j^2 = d(l_j^{t'}, p_{j,1}) + \sum_{l=1}^{|P_j^{t'}|-1} d(p_{j,l}, p_{j,l+1}), p_{j,l} \in P_j^{t'}, p_{j,l} \in \mathcal{T} \quad (2)$$

The flight distance of U_j between the l-th and $(l+1)$-th tasks is denoted by $d(p_{j,l}, p_{j,l+1})$. To make calculations easier, the position l_j^t of U_j at time t is considered as a virtual task point inserted into the sequence. This helps determine the flight distance of U_j from the last completed task point to the next pending task point during replanning initiation. Hence, the objective is:

$$\max \sum_{i=1}^{N_T} v_i - \sum_{j=1}^{N_U} \delta(D_j^1 + D_j^2) \tag{3}$$

Where δ is the weight factor denoting the fuel consumption per unit distance flown by the UAVs. The constraint of the model are as follows:

$$\delta(D_j^1 + D_j^2) + \sum_{l=1}^{|p_j|} e_{j,l} \leqslant e_j^0, U_j \in \mathcal{U} \tag{4}$$

Formula 4 denotes that the total energy consumed by U_j during the flight and the energy required to complete tasks should be less than its initial energy.

3 Methodology

In this section, a centralized-distributed negotiation-based incremental task replanning algorithm is designed for multi-UAV.

3.1 UAV Periodic Communication

During the task execution, UAVs require stable information sharing. Hence, we devised a periodic communication mechanism for achieving state awareness and information sharing among multi-UAVs. Define the concept of a package.

Definition 1. *At time t, U_j sends the package M_j^{PIT}:*

$$M_j^{PIT} = ID_j \parallel e_j^t \parallel P_j^t \parallel l_j^t \parallel t \parallel ADI \parallel PIT \tag{5}$$

Table 1. Additional information and types of packages

Number	Type	ADI	PIT
1	Handshake	/	“$hello$”
2	Help	l_i(Unprocessable task coordinates)	“sos”
3	Reply	$ID_{j'}$(Receiver identifier)	“$return$”
4	Search	ID_{lost}(Damaged UAVs identifier)	“$find$”

It contains self-identifier ID_j, remaining energy e_j^t, current coordinates l_j^t, unfinished task sequence P_j^t, current time t, additional information ADI and the type of package PIT. The specific contents of ADI and PIT are shown in Table 1. When UAV receives a package from another UAV, it first decodes the package tail to determine the type of the package.

Each U_j broadcasts M_j^{hello} to other UAVs at regular intervals during the task execution. At the same time, U_j also receives reply $M_{j'}^{hello}$ sent by $U_{j'}$. If no reply is received after U_j sends M_j^{hello}, it indicates that the receiver UAV damaged.

The periodic handshaking communication ensure that UAVs can achieve real-time perception and information sharing, unfinished task sequences can be backed up on other UAVs, and damaged UAVs can be detected in timely manner.

3.2 Task Replanning in the Case of New Tasks Sets

This section presents a task replanning algorithm for new task sets, which consists of three stages: Single UAV replanning, incremental replanning, and global replanning. **Algorithm 1** shows the specific process. The following provides a detailed exposition of the steps along with a thorough analysis of Figure 1.

Algorithm 1 Distributed Negotiation Incremental Algorithm

Input: The attributes of U_j, new task set $\mathcal{T}'$

Output: The task sequence $P_{j_k}^{t'}$ of U_{j_k} at t' or the task sequence $P_j^{t'}$ of U_j at t'

1: U_j uses single UAV replanning
2: **if** single UAV replanning has solution **then**
3:     Go to step 16
4: **end if**
5: U_j sends M_j^{sos}
6: The first $k = 1$ responding UAVs form $\mathcal{C}$ and uses incremental replanning
7: **if** current k-value incremental replanning algorithm has solution **then**
8:     Go to step 16
9: **end if**
10: The first $k + 1$ responding UAVs form $\mathcal{C}$ and uses incremental replanning
11: **if** current k-value incremental replanning algorithm has solution **then**
12:     Go to step 16
13: **end if**
14: U_j sends global help signal to BS
15: BS uses global replanning
16: **return** $P_{j_k}^{t'}$ or $P_j^{t'}$

As shown in Figure 1, at time $t = 0$, each UAV is executing its preplanned task sequence. At time t', U_j detects new task set $\mathcal{T}'$ (lower left), then U_j uses single UAV replanning to adjust P_j^t to accommodate $\mathcal{T}'$ (Module 1). If single UAV replanning is successful, then $P_j^{t'}$ is updated (Lines 1-4). Otherwise, U_j sends M_j^{sos} to other UAVs (Lines 5). $U_{j'}$ that receives M_j^{sos} sends $M_{j'}^{return}$ to U_j. U_j selects the first $k = 1$ ($k = 1, 2, 3 \dots$) UAVs that send reply package as members of $\mathcal{C}$ (Lines 6). Define the concept of temporary cluster.

Definition 2. *At time t, UAVs form temporary cluster $\mathcal{C}$:*

$$\mathcal{C} = \{U_{j_1}^{return}, U_{j_2}^{return}, \dots, U_{j_k}^{return} \mid U_{j_k}^{return} \in \mathcal{U}\} \tag{6}$$

At time t', pending task set within the cluster is updated to $\mathcal{T}_{add}^{new} = \cup_{k=1}^{|\mathcal{C}|} P_{j_k}^{t'} \cup \mathcal{T}'$, and cluster head U_j uses incremental replanning (Module 2). If the replanning is successful, $\mathcal{T}_{add}^{new}$ is planned among the members of $\mathcal{C}$ (Lines 7-9).If can't find solution. U_j selects the first $k+1$ UAVs that sends reply packages as $\mathcal{C}$ (Module ②) and uses incremental replanning (Lines 10) again.

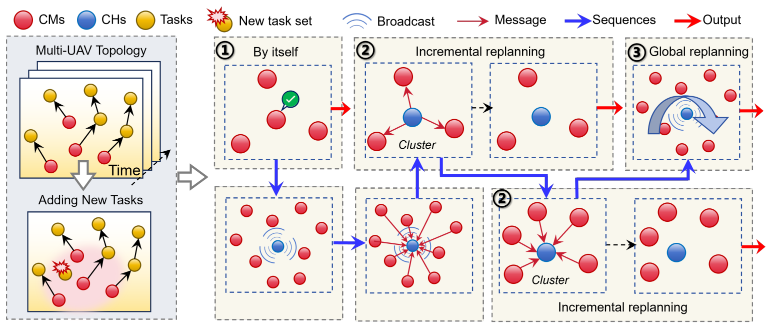

Fig. 1. Task replanning in the case of new task sets

If incremental replanning fails to output a path solution, U_j sends a global help signal to BS (Module ③). BS receives the signal and incorporates all the functioning UAVs into $\mathcal{C}$. Then it uses global replanning (Lines 15). All UAVs in the cluster execute their tasks in new sequence after successful global replanning.

3.3 Task Replanning in the Case of Damaged UAVs

In this section, based on the case of new task sets, we present task replanning algorithm for damaged UAVs.

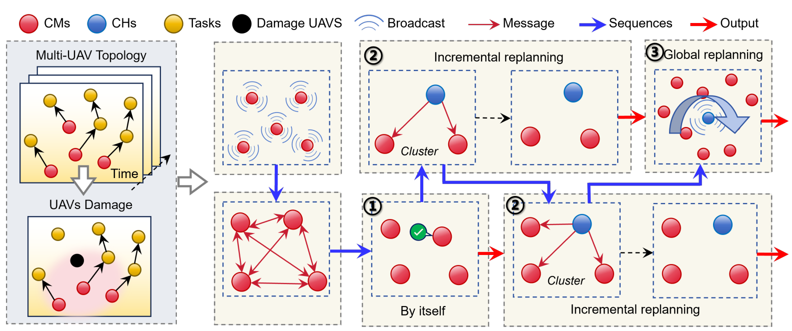

Fig. 2. Task replanning in the case of damaged UAVs

In Figure 2, if U_j does not receive M_{lost}^{hello} within the time interval at time t', it is considered that U_{lost} is damaged. Then, U_j sends M_j^{find} within its communication range. If $U_{j'}$ also does not receive M_{lost}^{hello}, it sends $M_{j'}^{return}$ to U_j.

Through the interaction of search packages, all UAVs that detect U_{lost} damaged as share information. the unfinished task of U_{lost} can be considered as new tasks, and the pending task set is updated to $\mathcal{T}_{lost}^{new} = \cup_{k=1}^{|\mathcal{C}|} P_{j_k}^{t'} \cup P_{lost}^{t'}$. The next steps are similar to new task set, but with one difference: the UAV closest to U_{lost} serves as cluster head, and the next k UAVs closest to U_{lost} serves as cluster members.

3.4 Task Reconstruction Algorithm

To solve the task planning and sequence problem for $\mathcal{T}^{new} \in \{\mathcal{T}^{new}_{lost}, \mathcal{T}^{new}_{add}\}$ in dynamic environments, we propose a task reconstruction algorithm to solve the problem, inspired by the literature[12], the algorithm consists of three steps, task redistribution, task reassignment, and task path optimization.

Algorithm 2 Optimization Algorithms For Energy Path Consideration

---

Input: The attributes of U_{j_k}, subtask set $\mathcal{T}^q_{sub}$ assigned to U_{j_k}, Maximum iterations number *max_gen*
Output: The task sequence $P^{t'}_{j_k}$ of U_{j_k} at time t'
1: *population* $\leftarrow \emptyset$, **random** $P^{t'}_{jk}$ ▷ Initialization stage
2: Calculated distance matrix for all tasks *dist_matrix*
3: **for** $gen = 1$ to *max_gen* **do**
4:     **for** *each indiv* $\in$ *population* **do** ▷ Fitness evaluation stage
5:         Calculate the energy required e_{plan} for the task sequence *indiv*
6:         **if** $e_{plan} > e^{t'}_j$ **then**
7:             $fitness \leftarrow 0$ ▷ The individual is invalid
8:         **else**
9:             Calculate *indiv* adaptability $fitness$
10:         **end if**
11:     **end for**
12:     Perform selection, crossover, and mutation genetic operators
13: **end for**
14: **return** $P^{t'}_{j_k}$ ▷ Output Stage

First, the task redistribution divides $\mathcal{T}^{new}$ into $|\mathcal{C}|$ subtask sets $\mathcal{T}^{new} = \{\mathcal{T}^1_{sub}, ..., \mathcal{T}^q_{sub}, ..., \mathcal{T}^{|\mathcal{C}|}_{sub}\}$, $\mathcal{T}^q_{sub} \subset \mathcal{T}^{new}$, through clustering algorithm[13], which clusters the tasks based on their coordinates. At the same time, the coordinates of the center point l^q_{sub} of each subtask set are recorded.

Then, the task assignment problem can be simplified to the bipartite graph assignment problem in the task reassignment, as the number of subtask centers and UAVs in $\mathcal{C}$ are equal. Specifically, U_{j_k} and $\mathcal{T}^q_{sub}$ are treated as two sets of vertices in the bipartite graph, and the cost matrix $\mathcal{D} = \{d(\mathcal{T}^q_{sub}, U_{j_k})\}_{|\mathcal{C}|\times|\mathcal{C}|}$ is formed, where $d(\mathcal{T}^q_{sub}, U_{j_k})$ denotes the distance between l^q_{sub} and l_{j_k}, $\mathcal{T}^q_{sub} \subset \mathcal{T}^{new}, U_{j_k} \in \mathcal{C}$. The final cost matrix, after subtracting rows and columns, has exactly one zero element in each row and column. These zero elements correspond to the task assignment with the minimum total cost.

Despite the task assignment being completed, the sequence of task execution within $\mathcal{T}^q_{sub}$ is still chaotic, so the final step is to optimize the path of the U_{j_k} assigned subtask set $\mathcal{T}^q_{sub}$ through optimization algorithms for energy path consideration, as shown in **Algorithm 2**.

The gene encoding the internal task sequence in $\mathcal{T}^q_{sub}$ is randomly sorted (Lines 1-2). The optimal chromosome is defined as the shortest path that requires energy no more than $e^{t'}_j$. The fitness function of the chromosome is:

$$e^{plan}_{j_k} = \sum_{l=1}^{|P_{j_k}|-1} \delta \cdot d\left(p_{j_k,l}, p_{j_k,l+1}\right) + \sum_{l=1}^{|P_{j_k}|} e_{j_k,l} \tag{7}$$

$$fitness = \begin{cases} 1/\sum_{l=1}^{|P_{j_k}|-1} d\left(p_{j_k,l}, p_{j_k,l+1}\right) & ,e_{j_k}^{plan} \le e_j^{t'} \\ 0 & ,e_{j_k}^{plan} > e_j^{t'} \end{cases} \tag{8}$$

Formula 7 denotes the energy required to complete the P_{j_k}, which consists of two parts: the product of path length and unit fuel consumption of U_{j_k} executing P_{j_k}, and the energy required to complete P_{j_k}. When $e_{j_k}^{plan} \le e_j^{t'}$, maximizing fitness means minimizing flight distance for optimal path planning. The chances of being chosen for further evolution are increased (Line 4-11). Through genetic operators (Line 12), the highest fitness $P_{j_k}^{t'}$ is ultimately outputted.

4 Experimental Results and Analysis

In this section, the effectiveness and superiority of the proposed ITRA are verified through simulation experiments.

4.1 Experimental Setup

Table 2. Experimental simulation parameters

Parameters	Numerical value	Parameters	Numerical value
number of tasks	$N_T = 28$	Initial number of UAVs	$N_U = 5$
Value of T_i	$T_i = 100$	Energy required to complete T_i	$e_i \in [10, 50]$
energy of U_j	$e_j^0 \in [500, 900]$	Energy consumption of UAV unit	$\delta = 1$

The proposed method is tested with 28 tasks uniformly distributed in a two-dimensional space of size 100 × 100. The task attributes have been obtained beforehand through other reconnaissance methods. Initially, 5 UAVs are located at coordinates (50, 0), and the results of the preplanning are shown in the second column of Table 3. The experimental code is written in Python 3.12[1]. Unless specified otherwise, the simulation parameters follow those shown in Table 2.

4.2 Simulation Results in the Case of New Task Sets

To verify the effectiveness of the algorithm in the case of new tasks, the simulation experiment is set at Time Step = 8, where U_3 scans new task set $\mathcal{T}'$=$[T_{29}, T_{30}, T_{31}]$, and then U_3 chooses to use the corresponding replanning algorithm based on whether e_3^8 can meet the needs of $\mathcal{T}_{add}^{new}$.

The single UAV replanning of U_3 is invoked when it has enough energy to meet the demands of $\mathcal{T}_{add}^{new}$. P_3^8 is updated with the inclusion of tasks T_{29}, T_{30}, T_{31}. After replanning, P_3^8 becomes $[T_{14}, T_{19}, T_7, T_{27}, T_{29}, T_{31}, T_{30}, T_{17}, T_{23}]$. Figure 3 illustrates the entire replanning process. It can be observed that when U_3 can handle the new task set on its own, other UAVs are not triggered for replanning.

[1] Our code for the paper is publicly available on https://github.com/Agentyzu/ITRA.

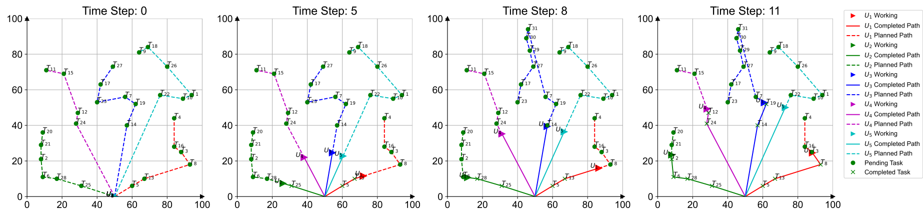

Fig. 3. Single UAV replanning diagram in the case of new task

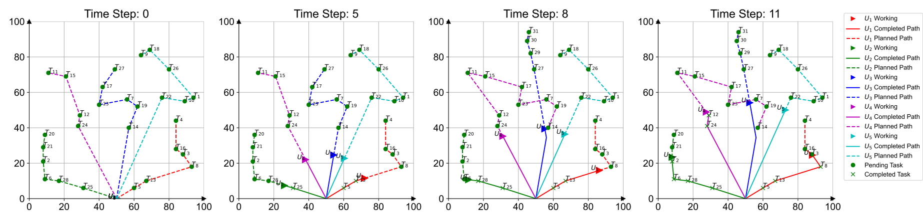

Fig. 4. Incremental replanning diagram in case of new task

When e_3^8 is insufficient to execute $\mathcal{T}_{add}^{new}$, the path diagram of the process is shown in Figure 4. The incremental replanning algorithm illustration demonstrates that task replanning within the cluster does not impact the task sequence of UAV outside the temporary cluster.

4.3 Simulation Results in Case of Damaged UAVs

To verify the effectiveness of ITRA in the case of damaged UAVs, experiment sets U_3 damage at time step $= 8$.

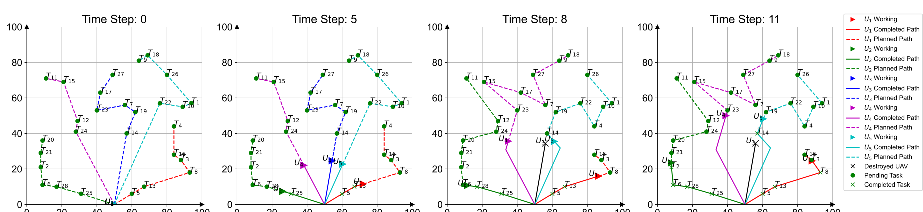

Fig. 5. Global replanning diagram in case of damaged UAVs

In this case, single UAV replanning and incremental replanning are similar to the case of new task sets, when the two replanning methods fail to produce a solution. The BS switches to global replanning and forms a temporary swarm of normally functioning UAVs. BS uses the task reconstruction algorithm to generate replanned task sequences for each UAV, as shown in the third column of the table 3 (completed tasks are highlighted in red). The entire process is shown in Figure 5.

Table 3. The task sequence for global replanning under damaged UAVs

UAV	Preplanned task sequence	Global replanning task sequence
U_1	$[T_5, T_{13}, T_8, T_3, T_{16}, T_4]$	$[T_5, T_{13}, T_8, T_3, T_{16}]$
U_2	$[T_{25}, T_{28}, T_6, T_2, T_{21}, T_{20}]$	$[T_{25}, T_{28}, T_6, T_2, T_{21}, T_{20}, T_{24}, T_{12}, T_{11}]$
U_3	$[T_{14}, T_{19}, T_7, T_{23}, T_{17}, T_{27}]$	/
U_4	$[T_{24}, T_{12}, T_{15}, T_{11}]$	$[T_{23}, T_{15}, T_{17}, T_7, T_{27}, T_9, T_{18}]$
U_5	$[T_{22}, T_{10}, T_1, T_{26}, T_{18}, T_9]$	$[T_{14}, T_{19}, T_{22}, T_4, T_{10}, T_1, T_{26}]$

4.4 Comparative Experimental Analysis

To verify the superiority of the algorithm, we compare ITRA with global replanning and evaluate the overall model utility. As depicted in Figure 6, we create four scenes by varying the number of new tasks and damaged UAVs, following the existing effectiveness experiment setup. Each scene generates 600 rounds of random tasks, and we calculate average utility values to derive final results.

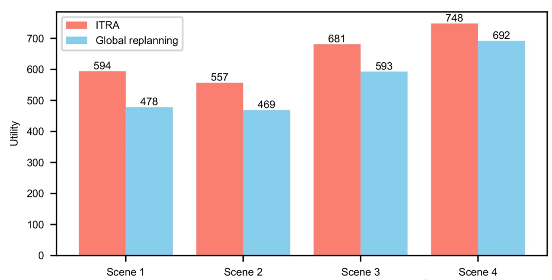

Fig. 6. Utility comparison diagram between ITRA and global replanning

The utility gain in ITRA is higher than global replanning, as demonstrated in Scene 1 where we compared three new tasks. In Scene 2, with an increase to five new tasks, the experimental results still show that ITRA has a higher utility compared to global replanning. When considering one damaged UAV (Scene 3), the experimental results indicate that ITRA achieves a higher utility gain than global replanning. Increasing the number of damaged UAVs to two (Scene 4), although there is a slight decrease in the utility gap between ITRA and global replanning, ITRA still outperforms global replanning in terms of utility.

5 Conclusion

The paper presents dynamic task replanning model for multi-UAV cooperative task execution, addressing dynamic environments like new tasks and damaged UAVs. A periodic communication mechanism is established for information sharing and autonomous regulation among UAVs. And a centralized-distributed negotiation algorithm is proposed for multi-UAV replanning. The algorithm details single UAV replanning, incremental replanning, and global replanning. Simulation experiments validate the effectiveness of the task replanning model in dynamic environments and demonstrate its superiority through experimental comparisons.

Through the experimental scenarios above, the ITRA can reduce the number of UAVs participating in the replanning in dynamic environments. When the uncompleted task set of the damaged UAVs can't be solved by the incremental replanning, the BS can use global planning.

References

1. Yang Xu et al. A hybrid algorithm based on mosfla and ga for multi-uavs plant protection task assignment and sequencing optimization. Applied Soft Computing, 96:106623, 2020.
2. Yang Li et al. Optimal spraying task assignment problem in crop protection with multi-uav systems and its order irrelevant enumeration solution. Biosystems Engineering, 214:177–192, 2022.
3. Ghosh, Saptarshi et al. Development of an iot based robust architecture for environmental monitoring using uav. In 2019 IEEE 16th India Council International Conference (INDICON), pages 1–4. IEEE, 2019.
4. Wang, Huifang et al. The use of unmanned aerial vehicle in military operations. In International Conference on Man-Machine-Environment System Engineering, pages 939–945. Springer, 2020.
5. Weinan Wu et al. Fast and coupled solution for cooperative mission planning of multiple heterogeneous unmanned aerial vehicles. Aerospace Science and Technology, 79:131–144, 2018.
6. Cui, Wei et al. Distributed task allocation for a multi-uav system with time window constraints. Drones, 6(9), 2022.
7. Hao CHEN et al. Formation flight of fixed-wing uav swarms: A group-based hierarchical approach. Chinese Journal of Aeronautics, 34(2):504–515, 2021.
8. Junwei et al. Study on multi-uav task clustering and task planning in cooperative reconnaissance. In 2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics, volume 2, pages 392–395, Aug 2014.
9. Yang Mi et al. A resource-constrained distributed task allocation method based on a two-stage coalition formation methodology for multi-uavs. The Journal of Supercomputing, 78(7):10025–10062, 2022.
10. Yang, Mi et al. A distributed task reassignment method in dynamic environment for multi-uav system. Applied Intelligence, 52(2):1582–1601, 2022.
11. Wu, Husheng et al. Modeling and simulation of dynamic ant colonys labor division for task allocation of uav swarm. Physica A: Statistical Mechanics and its Applications, 491:127–141, 2018.
12. Cristian Ramirez Atencia and et al. Weighted strategies to guide a multi-objective evolutionary algorithm for multi-uav mission planning. Swarm and Evolutionary Computation, 44:480–495, 2019.
13. Jinhua Xu et al. Collaborative orchard pesticide spraying routing problem with multi-vehicles supported multi-uavs. Journal of Cleaner Production, 458:142429, 2024.