

## Article

# An Event-Triggered Path-Based Bid Assignment Method for Energy-Aware Dynamic Multi-UAV Task Allocation

Yuhua Cong <sup>1,2</sup> , Yujia Li <sup>3</sup>, Xian Zhu <sup>2</sup> and Zhisheng Wang <sup>1,\*</sup><sup>1</sup> College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; congyuhua@nuaa.edu.cn<sup>2</sup> College of Computer and Artificial Intelligence, Nanjing University of Science and Technology Zijin College, Nanjing 210023, China<sup>3</sup> Commercial Aircraft Corporation of China, Beijing 102200, China

\* Correspondence: wangzhisheng@nuaa.edu.cn

## Highlights

### What are the main findings?

- ET-PBBA performs one-pass event-driven bid assignment on the active-task and available-UAV set, rather than multi-round bundle negotiation.
- Controlled paired-seed tests identify event timing and the path-dependent physical-cost block as the principal sources of improvement.

### What are the implications of the main findings?

- The method provides a lightweight event-local reassignment layer for obstacle-constrained, energy-limited, and time-sensitive UAV missions.
- The physical replay complements the allocation study with measured waypoint-tracking and translational-kinematics evidence from the available single-UAV platform.

## Abstract

This study addresses dynamic multi-UAV task allocation under online task release, static obstacles, deadlines, limited energy, and vehicle failures. We propose an event-triggered path-based bid assignment method (ET-PBBA). At each mission event, an A-star module estimates the static-obstacle travel cost of feasible UAV–task pairs, while a normalized score combines reward, path length, predicted energy, and secondary load–energy pressure. A one-pass sorted scan then produces an event-local one-to-one assignment. Paired-seed mechanism tests show that event-time activation raises completion rate by 0.046 in the main scenario and 0.065 under failure stress relative to 10 s periodic allocation. The path-dependent physical-cost block provides the dominant improvement over reward-only ranking, whereas the balance term is not significant in the ordinary scenarios. For 10 UAVs and 50 tasks, ET-PBBA reduces average completion time by 5.2% and 4.1% and total energy by 4.3% and 3.0% relative to auction and CBBA, respectively, although Hungarian remains the stronger centralized reference. Under failure stress, completion reaches 0.790 and deadline violations decrease to 9.73. Single-UAV replay completes all 50 nominal waypoints with 9.16 cm mean error, providing auxiliary tracking evidence rather than validation of nonlinear or online multi-UAV execution.

**Keywords:** multi-UAV task allocation; event-triggered reassignment; path-based bid assignment; energy-aware allocation



Academic Editor: Peihu Duan

Received: 18 June 2026

Revised: 24 July 2026

Accepted: 29 July 2026

Published: 9 September 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

Multi-UAV missions usually evolve under coupled spatial, temporal, and resource constraints. From the perspective of task allocation, they belong to the broader multi-robot task allocation family, where task structure, robot capability, and temporal coupling determine both model complexity and solution strategy [1]. Later taxonomies show that once tasks are no longer independent single-robot jobs, utility coupling and ordering constraints become central [2]. This issue is particularly relevant in UAV missions, where task release, service duration, deadlines, and disturbance recovery are often inseparable from the allocation itself [3]. Recent reviews likewise identify task allocation as a persistent bottleneck in practical UAV teams [4]. The difficulty becomes sharper in dynamic settings, where task arrivals, vehicle failures, deadline pressure, and residual-energy variation appear after the initial plan has already been issued [5]. Communication and energy limitations further constrain what can be done online [6], and trajectory-level studies make clear that motion and energy expenditure are already coupled before assignment is finalized [7].

Classical and distributed methods provide the main reference points, but their assumptions are not equally suitable for online UAV execution. The Hungarian method remains attractive when a complete cost matrix and global state are available, yet this centralized setting becomes restrictive when vehicles fail or new tasks appear after takeoff [8]. Auction algorithms reduce the need for a central coordinator, although their performance depends strongly on whether the bid utility contains the physical cost that later determines the assignment [9]. Market-based robot coordination has therefore remained useful in practice, but it also shows that the market rule alone does not define the quality of the allocation [10]. Consensus-based bundle allocation extends auctions to ordered task bundles and distributed agreement, making CBBA a natural benchmark for multi-agent missions [11]. However, the standard CBBA mechanism is not designed to recompute obstacle-aware travel cost and residual-energy pressure at every disturbance epoch. Auctioned POMDPs can represent uncertainty more explicitly, but the resulting computational burden is difficult to reconcile with frequent onboard reassignment [12]. Recent multi-criterion studies further confirm that dynamic allocation cannot be reduced to geometric distance alone [13]. A unified comparison of centralized and distributed multi-UAV allocation methods reaches a similar conclusion: robustness, scalability, and communication burden must be evaluated together rather than treated as separate after-the-fact metrics [14].

Optimization and learning approaches address the same problem from a different direction. Grey-wolf optimization has been adapted to heterogeneous UAV allocation under uncertainty, but population-based search is usually more suitable for planning-stage optimization than for frequent local repair during execution [15]. Improved A-QCDPSO combines particle-quality clustering with auction-based reassignment, although its iterative search still increases the computational burden when the active task set changes repeatedly [16]. ISOM with attention mechanisms introduces learned weighting into collaborative allocation and improves representational flexibility, but the learned policy is less transparent than an explicit bid rule when a mission operator needs to understand why ownership changes [17]. Multi-agent reinforcement learning can coordinate heterogeneous UAV teams, yet its behavior remains tied to training coverage and deployment mismatch [18]. Event-triggered graph reinforcement learning explicitly considers weak communication, but it still depends on offline policy learning and does not directly provide the simple paired cost terms needed for auditable online reassignment [19]. These studies motivate the present choice of a deterministic event-driven heuristic rather than a heavier global optimizer or a black-box learned policy.

Operational constraints motivate a complementary line of work. In search-and-rescue applications, for example, allocation objectives vary with payload, vehicle role, and mission urgency [20]. Communication-constrained methods restrict the information that can be exchanged during consensus [21]. Clustering-based strategies reduce negotiation scope to save communication effort [22]. Resource-constrained coalition formation addresses tasks that require coordinated UAV groups rather than single-vehicle assignment [23]. Contract-net variants provide lightweight coordination mechanisms for heterogeneous UAV swarms [24], and two-stage reassignment schemes show how disturbance recovery can be organized without rebuilding the whole mission plan [25].

Dynamic missions therefore call for repair rather than repeated full replanning. Event-local reassignment can restore feasibility without discarding the entire allocation, but a repair rule is only useful if it is triggered by changes that matter to the executing team [26]. Interleaved allocation, scheduling, and motion planning show that recovery is stronger when assignment and path feasibility are considered together [27]. Event-triggered allocation follows the same logic by activating reassignment only when feasibility or utility has materially changed [28]. The path layer is part of this decision rather than a downstream visualization step. An  $A^*$ -based shortest-path module can replace Euclidean distance with a static-obstacle-aware travel estimate [29]. Incremental replanning studies further show that online updates benefit from reusing previous search information, which is consistent with local repair rather than repeated full optimization [30]. Multi-agent pathfinding highlights an additional difficulty, namely that simultaneous routes can be coupled even when tasks appear separable at the assignment layer [31]. Recent studies on dense multi-vehicle planning, cluttered workspaces, cramped multi-UAV environments, and joint task-allocation/trajectory design all point to the same practical limitation: route feasibility can change which UAV should own a task, but this information is often not fed back into a lightweight event-time bid in a transparent form [32–35].

The literature therefore points to a narrower gap in online execution. When tasks appear during the mission, when deadlines become tight, or when a UAV fails, the allocator must respond at the event time without rebuilding the entire mission plan from scratch. Existing centralized methods provide useful references but assume global state. Standard auction and CBBA variants provide natural comparison points, yet their ranking rules do not necessarily expose how obstacle-aware travel and residual-energy pressure affect each event-local decision. Learning-based methods can adapt to richer state descriptions, but they make the allocation logic less transparent and require training coverage that may not match the disturbance pattern at deployment. The resulting design question is how to combine static-obstacle path cost and energy pressure in an auditable bid rule that remains simple enough for repeated use on the currently active tasks and available UAVs. ET-PBBA addresses this gap through a stage-wise reassignment rule evaluated only at event epochs. It applies an  $A^*$ -based path-cost module to every feasible UAV–task pair and weights task reward against path length, predicted energy use, and load and residual-energy pressure. The method is not intended to replace low-level trajectory tracking, centralized global optimization, CBBA-style bundle insertion, or packet-level distributed consensus. Its role is to provide a practical event-local reassignment layer for dynamic missions.

The contributions of this work are summarized as follows.

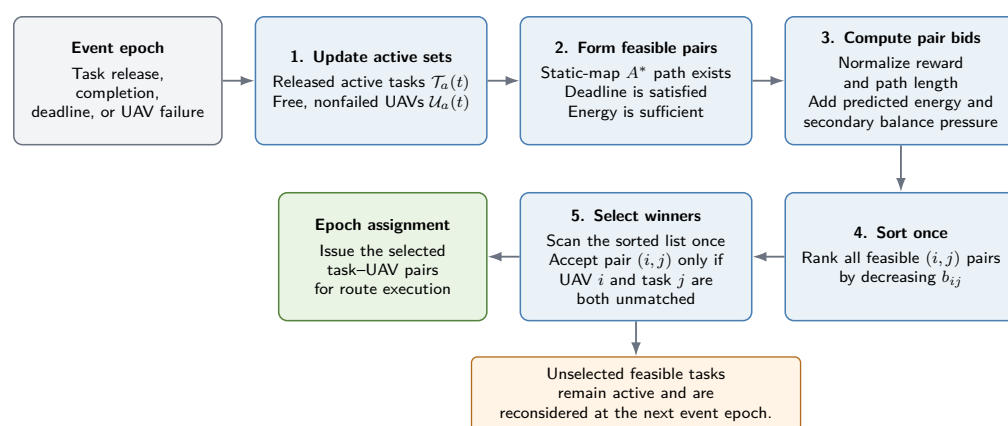
- An event-driven reassignment formulation is developed for dynamic multi-UAV missions. The decision process is activated only at task-release, task-completion, deadline, and UAV-failure epochs, and each allocation call is restricted to the currently active tasks and currently available UAVs.
- A path-based weighted bidding rule is implemented for one-pass event-local assignment. Each feasible UAV–task pair is scored by static-obstacle-aware  $A^*$  path length,

predicted energy consumption, and load and residual-energy pressure, so the mathematical score and winner-selection rule correspond directly to the simulation code.

- A controlled paired-seed study separates the effects of event timing, path-dependent physical cost, and energy balancing. It shows that event-time evaluation and the physical-cost block drive the repeatable gains, while the balance term acts only as a secondary regularizer in the tested energy-heterogeneous setting. Comparisons with auction and CBBA use the same exogenous events and pairwise feasibility interface.
- Simulation results cover dynamic task release, UAV failures, obstacle-density changes, and scalability. A separate single-UAV route-replay experiment reports waypoint-tracking accuracy and measured translational kinematics, without being used as evidence for nonlinear dynamic feasibility or online multi-UAV allocation.

## 2. Problem Formulation and Method

Figure 1 summarizes the implemented ET-PBBA decision logic. When a task-release, task-completion, deadline, or UAV-failure event occurs, the allocator is called on the currently active tasks and the currently available UAVs. Each feasible UAV–task pair is scored once by a path-based weighted bid, and a greedy nonconflicting winner scan produces a one-to-one assignment for that epoch. The analysis in this section is therefore written for the implemented bid-assignment process, not for a full bundle-consensus auction.



**Figure 1.** Event-triggered path-based bid-assignment mechanism of ET-PBBA.

### 2.1. Mission Model

The mission is modeled on a bounded two-dimensional workspace with known static obstacles. Map construction and low-level flight control are outside the allocation layer. The simulated workspace is a  $100 \times 100$  m square discretized into a  $40 \times 40$  occupancy grid. The path planner is used as a cost module that returns obstacle-aware travel time, energy cost, and feasibility for a candidate assignment. Let

$$\mathcal{U} = \{1, 2, \dots, N\} \quad (1)$$

denote the UAV set and

$$\mathcal{T}(t) = \{1, 2, \dots, M(t)\} \quad (2)$$

denote the tasks available at time  $t$ . UAV  $i$  has position  $p_i(t)$ , residual energy  $e_i(t)$ , and cruise speed  $v_i$ . All simulated UAVs use  $v_i = 1$  and default initial energy 220 unless a stress or balance scenario imposes a different initialization. Task  $j$  has location  $l_j$ , release time  $r_j$ , deadline  $d_j$ , service time  $s_j$ , and priority reward  $w_j$ . Tasks are indivisible and non-preemptive. A task that violates energy, deadline, or path-feasibility constraints is left unassigned and counted as incomplete in the evaluation.

To couple allocation with static-obstacle travel cost, the path planner is abstracted as a reusable cost interface. The path-cost module is denoted by

$$\rho(q, q') = (\tau(q, q'), \varepsilon(q, q'), \chi(q, q')), \quad (3)$$

where  $\tau(q, q')$  is the obstacle-aware travel time from position  $q$  to  $q'$ ,  $\varepsilon(q, q')$  is the corresponding energy cost, and  $\chi(q, q') \in \{0, 1\}$  indicates whether a static-obstacle-free grid path exists. The implementation obtains  $\tau$  from the  $A^*$  path length divided by UAV speed and computes  $\varepsilon$  as path length plus  $0.2s_j$  for service. This module checks static-obstacle feasibility for one UAV at a time and does not solve time-expanded inter-UAV collision avoidance. For a candidate pair  $(i, j)$  evaluated at event time  $t$ , define the predicted arrival time, finish time, and energy requirement as

$$a_{ij}(t) = \max\{r_j, t + \tau(p_i(t), l_j)\}, \quad (4)$$

$$f_{ij}(t) = a_{ij}(t) + s_j, \quad (5)$$

$$E_{ij}(t) = \varepsilon(p_i(t), l_j). \quad (6)$$

These quantities are evaluated from the UAV's current state and the candidate task only. They therefore match the one-pair-at-a-time decision implemented at each event epoch; no ordered task bundle or route-sequence optimizer is assumed.

Dynamic events are represented as

$$\mathcal{E}(t) = \{e_{\text{new}}, e_{\text{cancel}}, e_{\text{fail}}, e_{\text{energy}}, e_{\text{link}}\}, \quad (7)$$

corresponding to task arrival, task cancellation, UAV failure, energy infeasibility, and link-state change. The reported simulator calls the allocator at task-release, task-completion, deadline, and UAV-failure epochs. Task cancellation and link-state events remain part of the conceptual model but are not activated in the present experiments.

## 2.2. Event-Triggered Active-Set Update

ET-PBBA is implemented as an event-local bid-assignment heuristic. It is activated when a task is released, a task is completed, a deadline update is processed, or a UAV failure is detected. The algorithm does not run a multi-round consensus process inside an event epoch. Instead, each event defines the active tasks and available UAVs that are eligible for one reassignment call. For an event at time  $t$ , the active task set contains released tasks that have not been completed, canceled, or declared failed by the deadline rule. Canceled tasks are removed from the previous allocation and are not eligible for rebidding. Failed-UAV tasks return to the active pool if their deadlines and path feasibility still allow reassignment. The set used by the implemented runner is

$$\mathcal{T}_a(t) = \{j \in \mathcal{T}(t) : r_j \leq t, j \text{ is active}, t < d_j\}. \quad (8)$$

The available UAV set contains only vehicles that are nonfailed, currently free, and have positive residual energy,

$$\mathcal{U}_a(t) = \{i \in \mathcal{U} : i \text{ is nonfailed, free, and } e_i(t) > 0\}. \quad (9)$$

The current implementation therefore performs event-local reassignment over  $\mathcal{U}_a(t) \times \mathcal{T}_a(t)$ . This is narrower than a full local-bundle repair protocol, but it is exactly the decision rule used to produce all reported simulation results. Within this product set, a

candidate pair is admitted only after static-map path feasibility, deadline feasibility, and energy feasibility have been checked. The feasible candidate pool is

$$\mathcal{C}(t) = \{(i, j) \in \mathcal{U}_a(t) \times \mathcal{T}_a(t) : \chi(p_i(t), l_j) = 1, f_{ij}(t) \leq d_j, E_{ij}(t) \leq e_i(t)\}, \quad (10)$$

If  $(i, j) \notin \mathcal{C}(t)$ , the pair is not scored and cannot be selected in the current epoch.

### 2.3. Event-Local Assignment Objective

The optimization object associated with one event epoch is a bipartite one-to-one matching over the feasible candidate pool, rather than an ordered bundle-allocation problem. Let  $x_{ij}(t) = 1$  if feasible pair  $(i, j)$  is accepted at time  $t$ , and let  $x_{ij}(t) = 0$  otherwise. The event-local feasible set is

$$\mathcal{X}(t) = \left\{ x : x_{ij}(t) \in \{0, 1\}, \quad (i, j) \in \mathcal{C}(t), \right. \\ \sum_{j: (i, j) \in \mathcal{C}(t)} x_{ij}(t) \leq 1, \quad \forall i \in \mathcal{U}_a(t), \\ \left. \sum_{i: (i, j) \in \mathcal{C}(t)} x_{ij}(t) \leq 1, \quad \forall j \in \mathcal{T}_a(t) \right\}. \quad (11)$$

Using the pair score  $b_{ij}(t)$  defined below, the corresponding maximum-weight matching objective is

$$\max_{x \in \mathcal{X}(t)} J_t(x) = \sum_{(i, j) \in \mathcal{C}(t)} b_{ij}(t) x_{ij}(t). \quad (12)$$

This objective states the decision problem represented by the bid list. The implemented ET-PBBA routine does not solve this binary program exactly; it sorts the same pair scores once and returns a feasible member of  $\mathcal{X}(t)$  through the nonconflicting scan in Section 2.5. Consequently, the mathematical model, bid construction, and implemented winner-selection rule all operate on the same event-local pair set.

### 2.4. Path-Based Bidding with Secondary Balance Regularization

The bid is computed directly for each feasible UAV–task pair, which makes the implemented algorithm consistent with the simulation runner. Let  $L_{ij}$  denote the  $A^*$  path length from the current UAV position to task  $j$ , and let  $E_{ij} = \varepsilon(p_i(t), l_j)$  be the corresponding predicted travel-plus-service energy. The deadline is enforced by the feasibility condition above rather than by a separate soft risk term in the reported runner.

For any candidate reward or cost term  $z$ , the normalized value within the current feasible pool is

$$\hat{z} = \frac{z - z_{\min}}{z_{\max} - z_{\min} + \epsilon}, \quad (13)$$

where  $z_{\min}$  and  $z_{\max}$  are taken over  $\mathcal{C}(t)$ , and  $\epsilon > 0$  prevents division by zero. The load–energy balance pressure of UAV  $i$  is computed as

$$Q_i = \frac{1}{2} \hat{n}_i + \frac{1}{2} \hat{d}_i^e, \quad (14)$$

where  $\hat{n}_i$  is the normalized completed-task load and  $\hat{d}_i^e$  is the normalized residual-energy deficit. The implemented ET-PBBA score is therefore

$$b_{ij} = \hat{w}_j - 1.0 \hat{L}_{ij} - 0.4 \hat{E}_{ij} - 0.45 Q_i. \quad (15)$$

These coefficients are fixed across all reported scenarios. They were chosen after normalization to keep obstacle-aware travel cost dominant and energy and balance pressure

secondary. No scenario-specific tuning is used. Section 3.5 examines local coefficient perturbations.

### 2.5. Greedy Bid Selection and Conflict Resolution

After the feasible candidate pool has been scored, ET-PBBA sorts all pairs in descending order of  $b_{ij}$ , and the winner-selection step scans the sorted list once. A pair is accepted only if neither UAV  $i$  nor task  $j$  has already been selected in the current epoch; once accepted, that UAV and task are removed from further consideration in the same epoch. This update rule produces a one-to-one assignment without bundle insertion, task-order reinsertion, or multi-round negotiation. If no feasible pair exists for a task, it remains active until a later event or until its deadline is exceeded. The acronym ET-PBBA therefore denotes event-triggered path-based bid assignment, namely a one-pass event-local heuristic rather than a full bundle auction.

### 2.6. Analysis of ET-PBBA

The following analysis concerns the event-driven implementation that generates the reported experiments. It does not imply that the greedy scan solves the event-local maximum-weight matching problem exactly. The statements establish only the properties supported by Algorithm 1.

---

#### Algorithm 1 Event-local ET-PBBA execution

---

1. Receive a task-release, task-completion, deadline, or UAV-failure event at time  $t$ .
  2. Construct  $\mathcal{T}_a(t)$  and  $\mathcal{U}_a(t)$  from the updated mission state.
  3. Evaluate  $A^*$  path length, finish time, and energy for every pair in  $\mathcal{U}_a(t) \times \mathcal{T}_a(t)$ ; retain only  $\mathcal{C}(t)$ .
  4. Normalize reward, path length, energy, completed-task load, and residual-energy deficit over the current candidate pool.
  5. Compute  $b_{ij}$  for each  $(i, j) \in \mathcal{C}(t)$  and sort the pairs once in descending score order.
  6. Scan the sorted list and accept  $(i, j)$  only when neither  $i$  nor  $j$  has already been selected in the current epoch.
  7. Return the accepted one-to-one assignments; leave unselected feasible tasks active for a later event.
- 

**Proposition 1** (Feasibility of returned assignments). *Assume that the path-cost module reports infeasible path segments conservatively and that only candidate pairs satisfying the explicit energy and deadline checks are admitted to the bid list. Then every task that remains assigned after one ET-PBBA event epoch satisfies the assignment uniqueness, energy, deadline, and path-feasibility constraints.*

**Proof.** Assignment uniqueness follows from the greedy winner-selection step, which accepts a pair only if neither the UAV nor the task has already been used in the current epoch. Path feasibility is enforced by requiring finite  $A^*$  path length before a pair enters the candidate list. Energy and deadline feasibility are checked explicitly from the predicted travel-plus-service cost of that pair. Therefore, every admitted pair that survives the winner-selection step satisfies the stated constraints.  $\square$

**Proposition 2** (Bounded termination). *For a fixed event epoch with finite active-task and available-UAV sets, the implemented ET-PBBA call terminates after finitely many bid evaluations and one finite sorting step.*

**Proof.** At one event epoch, the algorithm enumerates all feasible UAV–task pairs drawn from finite active-task and available-UAV sets. Each pair receives one finite bid value. The

algorithm then sorts the finite candidate list and scans it once to select nonconflicting pairs. Every step therefore has finite cost, and the assignment call terminates.  $\square$

**Proposition 3** (Event-local decision scope). *At any event epoch, the implemented ET-PBBA call depends only on the tasks whose status is active and the UAVs that are both nonfailed and currently free.*

**Proof.** The candidate list is constructed only from active tasks and available UAVs. Tasks that are not yet released, already completed, declared failed by the deadline rule, or currently assigned are not scored at that epoch. UAVs that have failed, are busy executing a task, or have depleted energy are also excluded. Hence the output of the current allocation call depends only on the event-local active set.  $\square$

The propositions above characterize the reported ET-PBBA implementation as a bounded event-driven reassignment heuristic rather than an exact maximum-weight matching solver. It returns a feasible point in  $\mathcal{X}(t)$  and terminates in finite time, but it does not guarantee attainment of  $\max_{x \in \mathcal{X}(t)} J_t(x)$ . For the implemented epoch-wise assignment, the dominant cost is candidate evaluation. If the active-task set has size  $|\mathcal{T}_a|$  and the available-UAV set has size  $|\mathcal{U}_a|$ , the candidate-generation cost is

$$O(|\mathcal{U}_a||\mathcal{T}_a|C_p), \quad (16)$$

where  $C_p$  is the cost of one path-cost query. Sorting the feasible list adds  $O(K \log K)$ , where  $K \leq |\mathcal{U}_a||\mathcal{T}_a|$  is the number of feasible pairs. In the present simulator, the reported communication-related count is the number of feasible candidate pairs evaluated in one epoch. It is therefore renamed as the bid-pair count and should be interpreted as a bid-volume proxy rather than as packet-level communication traffic.

### 3. Simulation Results and Discussion

#### 3.1. Simulation Setup

The simulator was implemented in Python 3.14.5 by extending the public multi-UAV task allocation codebase in [14] with a dynamic event-driven runner. For a given paired seed, all algorithms use the same task set, UAV initial states, obstacle map, and exogenous task-release, failure, and deadline events. The environment is a  $100 \times 100$  m square with a  $40 \times 40$  occupancy grid. Obstacle cells are generated independently according to the prescribed density. UAV speed is fixed at 1, service time is sampled uniformly from 1 to 3, and the energy required to execute one task is modeled as path length plus 0.2 times service time. Path-aware travel cost is computed by a grid-based  $A^*$  planner, and all tables and figures in this section are generated from saved CSV logs. The  $A^*$  layer is used as a static-obstacle path-cost and feasibility interface. It does not model nonlinear flight dynamics, time-expanded inter-UAV conflict avoidance, wind, or actuator limits. A set of individually feasible grid paths should therefore be interpreted as an allocation-cost input, not as proof of dynamically feasible or simultaneous collision-free multi-UAV execution. The reported scripts were executed sequentially on an Intel Core i7-9700 processor at 3.00 GHz with 16 GB RAM; the measured runtime excludes parallel or GPU acceleration.

The external comparison uses the Hungarian, auction, and CBBA implementations from the cited codebase. Within each paired seed, all methods receive identical task releases, UAV failures, deadline updates, obstacle maps, and initial states. Task-completion events occur at method-dependent times because completion time is an algorithm outcome, but each completion invokes the same active-set update. The common runner supplies the same  $A^*$  path length, travel-plus-service energy model, deadline check, and residual-energy

feasibility check for every candidate pair; only the ranking and assignment rules differ. Thus the reported comparison does not attribute ET-PBBA's advantage to exclusive access to path or event information.

A separate controlled mechanism study isolates event timing and bid composition while holding the candidate pool and greedy selector fixed. It compares a 10 s periodic implementation of the full score, event-time reward-only ranking, event-time ranking with the path-length and path-dependent energy terms treated as one physical-cost block, and the full event-time score with balance pressure. Path length and energy are varied as one block because  $E_{ij} = L_{ij} + 0.2s_j$ ; deleting only  $\hat{L}_{ij}$  would leave the same path information in  $\hat{E}_{ij}$ . The score therefore contains only mechanisms that can affect the implemented active-set decision.

ET-PBBA evaluates all feasible pairs at each event epoch and uses the fixed coefficients (1.00, 0.40, 0.45) in every scenario. Main-scenario task releases are sampled over [0, 45], and deadlines equal release time plus an offset sampled over [80, 150]. In each failure-stress seed, two of the ten UAVs are selected uniformly without replacement, assigned initial energy 190, and given independent failure times sampled over [45, 95]. The balance-pressure scenario assigns initial energy in [125, 160] to the first third of UAVs and [225, 270] to the remainder, with tasks concentrated near the lower-energy group.

The recorded metrics are task completion rate, average completion time, makespan, total energy, residual-energy standard deviation, runtime, bid-pair count, and deadline violations. Results are reported as mean and standard deviation over paired seeds, and selected comparisons use paired Wilcoxon tests. The main, failure-stress, balance-pressure, and controlled mechanism studies each use 30 paired seeds. The dynamic-intensity and obstacle-density sweeps use 30 seeds per level. The scalability study spans 5/20, 10/50, 20/100, and 30/150 UAV–task settings with 30, 30, 30, and 10 seeds, respectively.

### 3.2. Main Dynamic Scenario

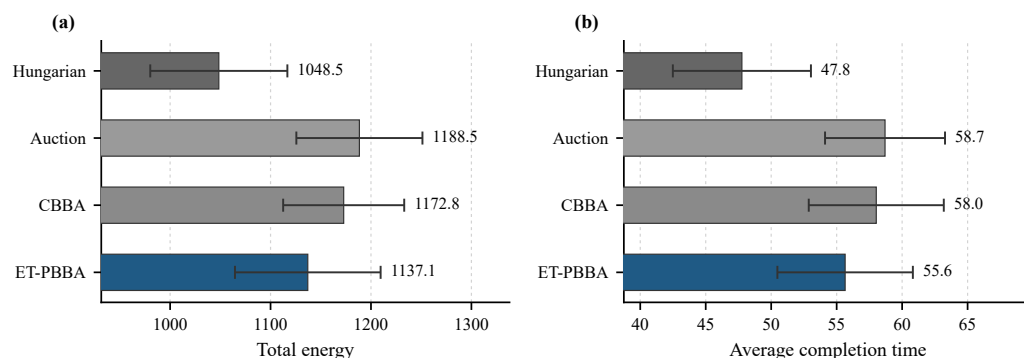
Table 1 reports the main dynamic scenario with 10 UAVs, 50 released tasks, and 10% obstacle density. All values are generated from 30 paired random seeds. ET-PBBA reaches a completion rate of  $0.916 \pm 0.047$ , compared with  $0.907 \pm 0.052$  for auction and  $0.907 \pm 0.052$  for CBBA. These completion-rate differences are not statistically significant: the paired Wilcoxon tests give  $p = 0.117$  against auction and  $p = 0.132$  against CBBA, with 95% confidence intervals of  $[-0.002, 0.019]$  and  $[-0.004, 0.021]$ . By contrast, ET-PBBA reduces average completion time by 5.2% against auction ( $p < 0.001$ ) and by 4.1% against CBBA ( $p < 0.001$ ). Total energy decreases by 4.3% ( $p < 0.001$ ) and 3.0% ( $p = 0.003$ ), respectively. The paired 95% confidence intervals are [1.95, 4.15] and [1.17, 3.59] for completion-time reduction, and [29.09, 73.68] and [13.72, 57.77] for energy reduction.

**Table 1.** Main dynamic scenario with 10 UAVs, 50 released tasks, and 10% obstacle density.

| Algorithm | Completion        | Avg. Time        | Energy            | Residual Energy Std. | Violations      | Bid Pairs        |
|-----------|-------------------|------------------|-------------------|----------------------|-----------------|------------------|
| Hungarian | $0.964 \pm 0.027$ | $47.76 \pm 5.27$ | $1048.5 \pm 68.3$ | $19.06 \pm 5.04$     | $1.73 \pm 1.28$ | $0.0 \pm 0.0$    |
| Auction   | $0.907 \pm 0.052$ | $58.69 \pm 4.58$ | $1188.5 \pm 62.8$ | $17.27 \pm 4.28$     | $4.50 \pm 2.52$ | $495.4 \pm 49.4$ |
| CBBA      | $0.907 \pm 0.052$ | $58.03 \pm 5.16$ | $1172.8 \pm 60.3$ | $16.35 \pm 3.86$     | $4.43 \pm 2.47$ | $503.8 \pm 51.9$ |
| ET-PBBA   | $0.916 \pm 0.047$ | $55.65 \pm 5.17$ | $1137.1 \pm 72.5$ | $15.58 \pm 3.39$     | $4.07 \pm 2.41$ | $505.8 \pm 48.3$ |

Figure 2 visualizes the statistically supported completion-time and energy reductions against auction and CBBA while making the comparison with Hungarian explicit. The centralized Hungarian assignment still achieves the highest completion rate together with the lowest completion time and total energy because it has access to a global solver. ET-PBBA is therefore not proposed as a replacement for exact centralized matching when a complete cost matrix and global state are continuously available. The evidence supports the narrower claim that its event-local path-based ranking is more effective than the tested

auction and CBBA implementations in this scenario. Because the present runner centrally constructs and sorts the candidate list, it does not establish a distributed implementation or a packet-level communication advantage.



**Figure 2.** Main scenario results (mean  $\pm$  SD over paired seeds): (a) total energy; (b) average completion time.

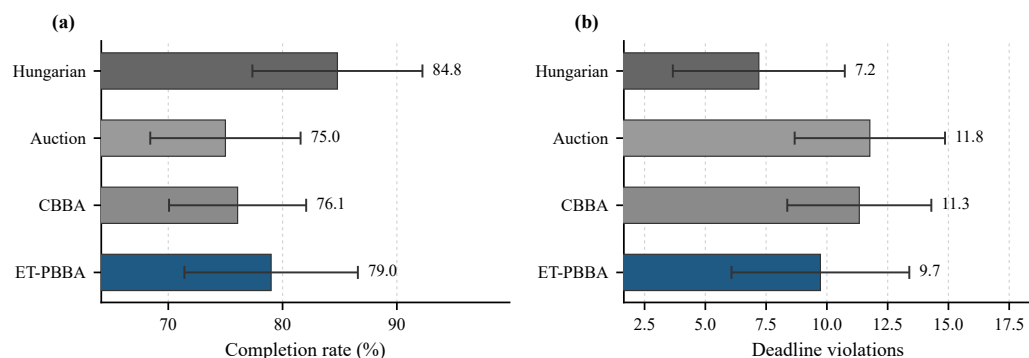
### 3.3. Failure Stress and Mechanism Isolation

Table 2 reports the failure-stress scenario. ET-PBBA raises completion rate from 0.750 to 0.790 relative to auction and from 0.761 to 0.790 relative to CBBA. Deadline violations decrease from 11.77 to 9.73 and from 11.33 to 9.73, respectively. The paired Wilcoxon tests support the completion-rate differences ( $p < 0.001$  and  $p = 0.005$ ) and the violation reductions ( $p < 0.001$  and  $p = 0.003$ ). Hungarian retains stronger absolute completion and deadline performance because it solves a centralized global assignment.

**Table 2.** Stress scenario with dense task release, 20% obstacle density, and UAV failures.

| Algorithm | Completion        | Avg. Time        | Energy           | Residual Energy Std. | Violations       | Bid Pairs        |
|-----------|-------------------|------------------|------------------|----------------------|------------------|------------------|
| Hungarian | $0.848 \pm 0.074$ | $44.53 \pm 4.37$ | $811.5 \pm 50.5$ | $16.62 \pm 4.58$     | $7.20 \pm 3.54$  | $0.0 \pm 0.0$    |
| Auction   | $0.750 \pm 0.066$ | $50.49 \pm 4.73$ | $872.1 \pm 56.1$ | $13.62 \pm 3.55$     | $11.77 \pm 3.09$ | $415.5 \pm 59.4$ |
| CBBA      | $0.761 \pm 0.060$ | $50.05 \pm 4.83$ | $869.6 \pm 60.3$ | $14.58 \pm 3.66$     | $11.33 \pm 2.96$ | $433.9 \pm 52.6$ |
| ET-PBBA   | $0.790 \pm 0.076$ | $49.81 \pm 3.98$ | $854.9 \pm 41.7$ | $13.59 \pm 3.48$     | $9.73 \pm 3.66$  | $433.5 \pm 63.2$ |

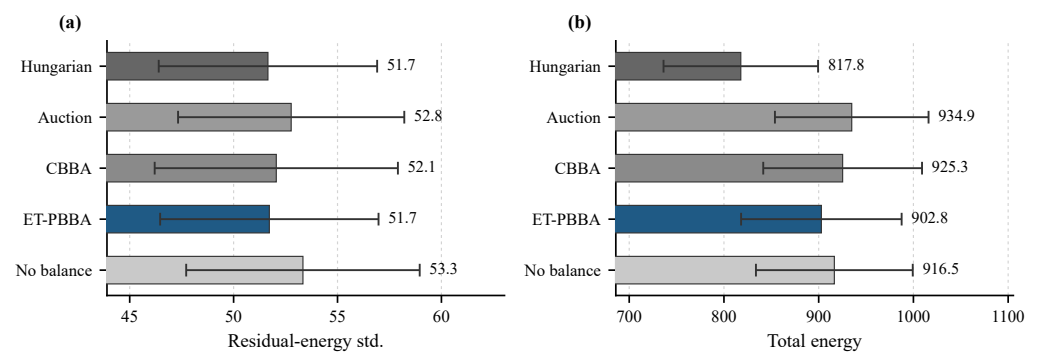
Figure 3 shows the same scenario from the two viewpoints that matter most after failures, namely how many tasks are eventually completed and how many miss their deadlines. The separation between ET-PBBA and the auction-type baselines becomes clearer here because failed-UAV tasks must be reassigned under reduced timing slack and tighter spatial constraints.



**Figure 3.** Failure-stress results (mean  $\pm$  SD over paired seeds): (a) completion rate; (b) deadline violations.

The balance-pressure scenario places more tasks near UAVs initialized with lower residual energy. In this case, the full ET-PBBA variant yields lower total energy than the

no-balance ablation (902.84 versus 916.53) together with a lower residual-energy standard deviation (51.72 versus 53.34). Relative to auction and CBBA, ET-PBBA also reduces energy from 934.9 and 925.3 to 902.8 while slightly reducing residual-energy dispersion. The full method therefore lies on the favorable side for both metrics when residual-energy heterogeneity is deliberately amplified, but the separation remains limited. The paired tests against the no-balance variant give  $p = 0.052$  for total energy and  $p = 0.078$  for residual-energy standard deviation, so the evidence is still short of a strong significance claim at the 0.05 level. Figure 4 presents the residual-energy dispersion and total-energy results.



**Figure 4.** Balance-pressure results for the balance-pressure scenario (mean  $\pm$  SD over paired seeds): (a) residual-energy standard deviation; (b) total energy.

Table 3 reports the controlled mechanism study. Event-time allocation increases completion rate by 0.046 in the main scenario and by 0.065 under failure stress relative to the 10 s periodic implementation of the same full score; both differences are significant ( $p < 0.001$ ). Deadline violations fall by 2.30 and 3.23 (both  $p < 0.001$ ). The main-scenario completion time is also 2.38 s lower ( $p = 0.006$ ), whereas the stress-scenario time difference is not significant ( $p = 0.516$ ). These comparisons isolate the benefit of allocating at mission events, but they do not explain the advantage over auction and CBBA because those baselines use the same event handler.

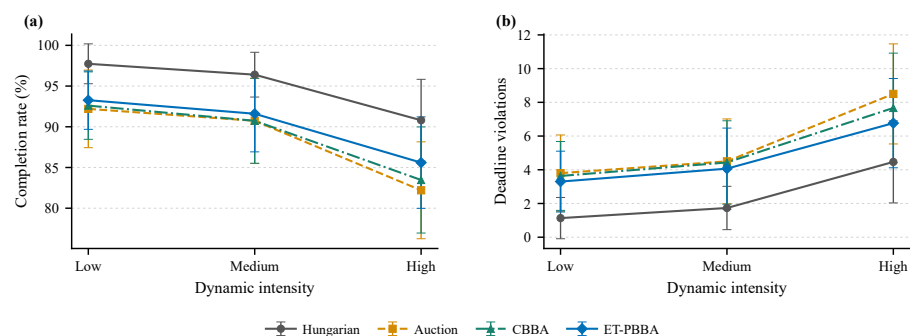
**Table 3.** Controlled mechanism study over 30 paired seeds. The physical-cost block contains the path-length and path-dependent energy terms.

| Scenario       | Variant               | Completion        | Avg. Time        | Energy/Task      | Violations       |
|----------------|-----------------------|-------------------|------------------|------------------|------------------|
| Main           | Periodic full (10 s)  | $0.850 \pm 0.039$ | $60.35 \pm 4.86$ | $22.87 \pm 1.95$ | $7.50 \pm 1.94$  |
|                | Event reward only     | $0.659 \pm 0.045$ | $82.75 \pm 5.80$ | $41.82 \pm 3.80$ | $17.07 \pm 2.23$ |
|                | Event + physical cost | $0.898 \pm 0.052$ | $57.79 \pm 5.27$ | $25.82 \pm 2.83$ | $5.10 \pm 2.60$  |
|                | Full ET-PBBA          | $0.896 \pm 0.049$ | $57.97 \pm 5.12$ | $25.91 \pm 2.67$ | $5.20 \pm 2.46$  |
| Failure stress | Periodic full (10 s)  | $0.695 \pm 0.058$ | $56.34 \pm 5.08$ | $23.29 \pm 2.48$ | $15.27 \pm 2.88$ |
|                | Event reward only     | $0.520 \pm 0.048$ | $76.24 \pm 5.13$ | $44.21 \pm 4.61$ | $24.00 \pm 2.41$ |
|                | Event + physical cost | $0.765 \pm 0.061$ | $55.89 \pm 6.33$ | $26.15 \pm 2.79$ | $11.73 \pm 3.03$ |
|                | Full ET-PBBA          | $0.759 \pm 0.063$ | $55.64 \pm 6.02$ | $26.33 \pm 3.05$ | $12.03 \pm 3.16$ |

Within the common event handler and greedy selector, the physical-cost block is the dominant source of improvement. Relative to reward-only ranking, it raises completion rate by 0.239 in the main scenario and by 0.245 under failure stress. It reduces average completion time by 24.96 and 20.36 s, energy per completed task by 16.00 and 18.06 units, and deadline violations by 11.97 and 12.27; every paired comparison gives  $p < 0.001$ . Adding balance pressure to the physical-cost score does not significantly change completion, time, energy, or violations in either ordinary scenario (all  $p \geq 0.280$ ). Together with the borderline balance-pressure results above, this evidence supports balance pressure only as a secondary energy-heterogeneity regularizer, not as the principal source of ET-PBBA's gains.

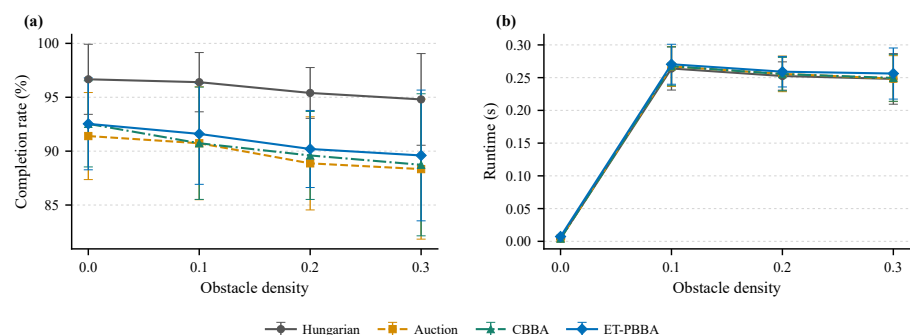
### 3.4. Dynamic Intensity, Obstacles, and Scalability

The dynamic-intensity sweep in Figure 5 shows a gradual degradation as task releases become denser. Completion rate decreases from 0.933 in the low-intensity setting to 0.856 in the high-intensity setting, while deadline violations increase from 3.30 to 6.77. The dominant effect is therefore a steady rise in time-window congestion rather than an abrupt collapse in completion, indicating that event-driven reassignment remains effective under sparse or moderate updates but becomes less flexible once many tasks compete for the same time windows.



**Figure 5.** Completion rate under different dynamic task-release intensities: (a) completion rate; (b) deadline violations.

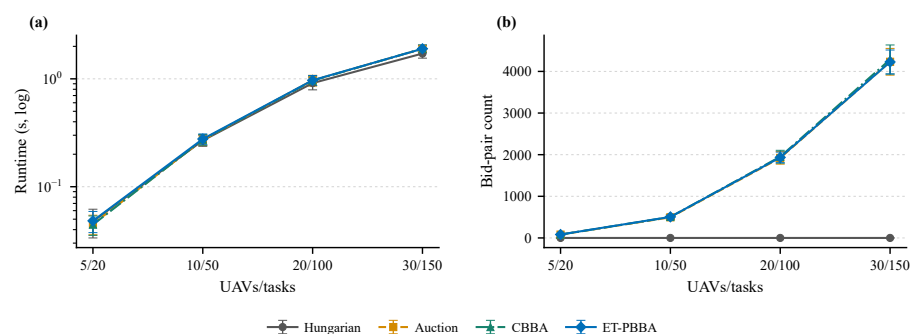
The obstacle-density sweep in Figure 6 exhibits a related tradeoff. ET-PBBA completion rate decreases moderately from 0.925 with no obstacles to 0.896 at 30% obstacle density, whereas runtime reacts much more strongly when obstacle-aware path queries replace direct Euclidean distance. The mean runtime is 0.007 s without obstacles and remains around 0.25–0.27 s at 10–30% obstacle density. Obstacle-aware path cost therefore makes static-grid feasibility explicit in cluttered maps, but it also becomes a dominant part of the computational burden.



**Figure 6.** Completion rate under different obstacle densities: (a) completion rate; (b) runtime.

Representative scalability results in Figure 7 show that ET-PBBA runtime grows from  $0.0483 \pm 0.0107$  s at 5 UAVs/20 tasks to  $1.8998 \pm 0.1581$  s at 30 UAVs/150 tasks, while the bid-pair count grows from  $82.63 \pm 12.78$  to  $4226.80 \pm 287.43$ . The corresponding runtime per evaluated pair decreases from  $0.599 \pm 0.174$  to  $0.452 \pm 0.052$  ms, indicating that fixed runner overhead does not explain the increase in total runtime. Completion rate increases from  $0.878 \pm 0.083$  to  $0.990 \pm 0.009$ , but this should not be interpreted as an algorithmic improvement with problem size. The workspace remains fixed at  $100 \times 100$  m while UAV count increases, so vehicle density and spatial coverage increase; the task-to-UAV ratio is also four at 5/20 and five at the other scales. This suite is therefore a computational scaling test rather than a constant-density workload test. Total runtime nevertheless grows by approximately 39 times and candidate volume by approximately 51 times. The

measurements do not establish a hard real-time guarantee; substantially larger missions would require candidate pruning, path-cost caching, or incremental path search, none of which is implemented here.



**Figure 7.** Runtime and bid-pair count across UAV–task sizes: (a) runtime; (b) bid-pair count.

### 3.5. Weight-Sensitivity Check

The fixed coefficients are further examined in an additional failure-stress suite over 30 paired seeds. Starting from (1.00, 0.40, 0.45), one coefficient is perturbed at a time while all scenario conditions remain paired. Table 4 reports changes relative to the baseline-weight result. Completion-rate changes lie between  $-0.010$  and  $+0.011$ , average completion-time changes between  $-0.70$  and  $+0.45$  s, and total-energy changes between  $-10.5$  and  $+8.9$  units. This is a local robustness check around the reported coefficients, not a claim of globally optimal tuning.

**Table 4.** Weight-sensitivity check in an additional failure-stress suite over 30 paired seeds.

| Variant      | Weights ( $\alpha_L, \alpha_E, \alpha_B$ ) | $\Delta$ Completion | $\Delta$ Avg. Time | $\Delta$ Energy |
|--------------|--------------------------------------------|---------------------|--------------------|-----------------|
| Base         | (1.00, 0.40, 0.45)                         | +0.000              | +0.00              | +0.0            |
| Path low     | (0.80, 0.40, 0.45)                         | $-0.010$            | +0.45              | +8.0            |
| Path high    | (1.20, 0.40, 0.45)                         | +0.011              | $-0.70$            | $-10.0$         |
| Energy low   | (1.00, 0.25, 0.45)                         | $-0.004$            | +0.40              | +8.9            |
| Energy high  | (1.00, 0.55, 0.45)                         | +0.003              | $-0.60$            | $-10.5$         |
| Balance low  | (1.00, 0.40, 0.25)                         | +0.000              | $-0.23$            | $-1.5$          |
| Balance high | (1.00, 0.40, 0.65)                         | +0.001              | $-0.45$            | $-3.8$          |

Table 5 summarizes the selected paired statistical comparisons after the scenario-level results. This placement separates the descriptive scenario tables and figures from the inferential effect estimates, and it highlights the uncertainty of the main completion-time, energy, completion-rate, and deadline-violation improvements.

**Table 5.** Selected paired comparisons with 95% confidence intervals (CIs) for paired-seed effects. Positive effects favor ET-PBBA; negative effects favor the comparison method.

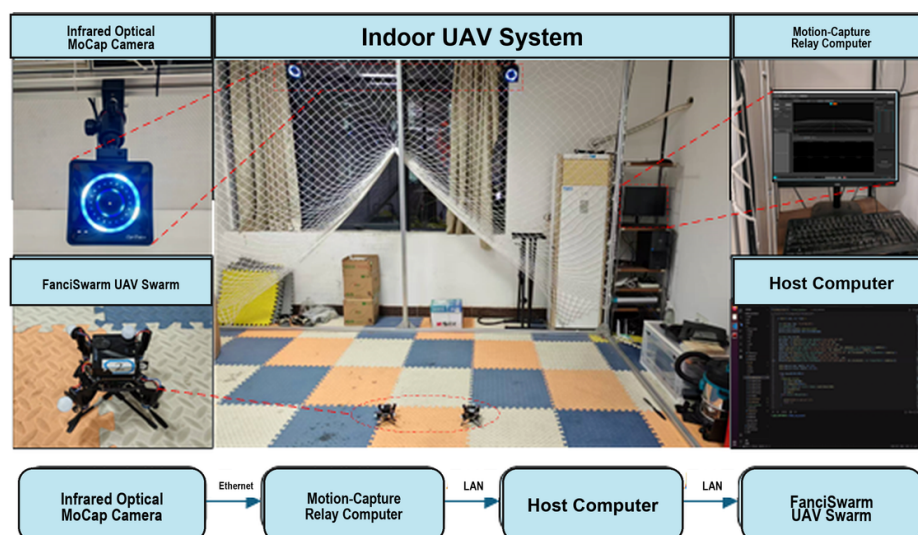
| Scenario | Comparison | Metric              | Effect   | 95% CI                  | Effect (%) | Better Seeds | p      |
|----------|------------|---------------------|----------|-------------------------|------------|--------------|--------|
| Main     | Auction    | avg completion time | 3.05     | [1.95, 4.15]            | 5.2        | 90.0%        | <0.001 |
| Main     | Auction    | total energy        | 51.38    | [29.09, 73.68]          | 4.3        | 83.3%        | <0.001 |
| Main     | Auction    | completion rate     | 0.009    | [ $-0.002$ , 0.019]     | 1.0        | 33.3%        | 0.117  |
| Main     | CBBA       | avg completion time | 2.38     | [1.17, 3.59]            | 4.1        | 73.3%        | <0.001 |
| Main     | CBBA       | total energy        | 35.74    | [13.72, 57.77]          | 3.0        | 73.3%        | 0.003  |
| Main     | CBBA       | completion rate     | 0.009    | [ $-0.004$ , 0.021]     | 1.0        | 40.0%        | 0.132  |
| Stress   | Auction    | completion rate     | 0.040    | [0.024, 0.056]          | 5.3        | 76.7%        | <0.001 |
| Stress   | Auction    | deadline violations | 2.03     | [1.21, 2.86]            | 17.3       | 80.0%        | <0.001 |
| Stress   | CBBA       | completion rate     | 0.029    | [0.009, 0.050]          | 3.9        | 63.3%        | 0.005  |
| Stress   | CBBA       | deadline violations | 1.60     | [0.62, 2.58]            | 14.1       | 63.3%        | 0.003  |
| Balance  | No balance | total energy        | 13.70    | [0.69, 26.70]           | 1.5        | 46.7%        | 0.052  |
| Balance  | No balance | residual energy std | 1.62     | [ $-0.08$ , 3.31]       | 3.0        | 50.0%        | 0.078  |
| Main     | Hungarian  | completion rate     | $-0.048$ | [ $-0.063$ , $-0.033$ ] | $-5.0$     | 6.7%         | <0.001 |
| Main     | Hungarian  | residual energy std | 3.48     | [1.01, 5.94]            | 18.3       | 73.3%        | 0.009  |

## 4. Physical Route-Replay Experiment

### *Hardware Platform and Replay Protocol*

The simulation results above evaluate allocation behavior. The physical experiment addresses only whether the generated routes can be replayed on the available UAV platform. Because the platform supports waypoint execution on one vehicle rather than synchronous deployment of the virtual multi-UAV team, the experiment uses repeated single-UAV replay. It does not evaluate online allocation, event-triggered reassignment, communication, or inter-UAV conflict handling. The acceptance threshold is 0.25 m in horizontal waypoint error.

Figure 8 presents the indoor hardware platform used for the replay experiments. The setup consists of an infrared optical motion-capture camera network, a motion-capture relay computer, a host computer, and the UAV execution platform. During replay, the motion-capture system provides real-time pose measurements, the relay computer forwards the measured state to the host computer, and the host computer issues waypoint-level commands to the UAV. This arrangement enables repeatable indoor execution and synchronized logging of the commanded route and measured trajectory.

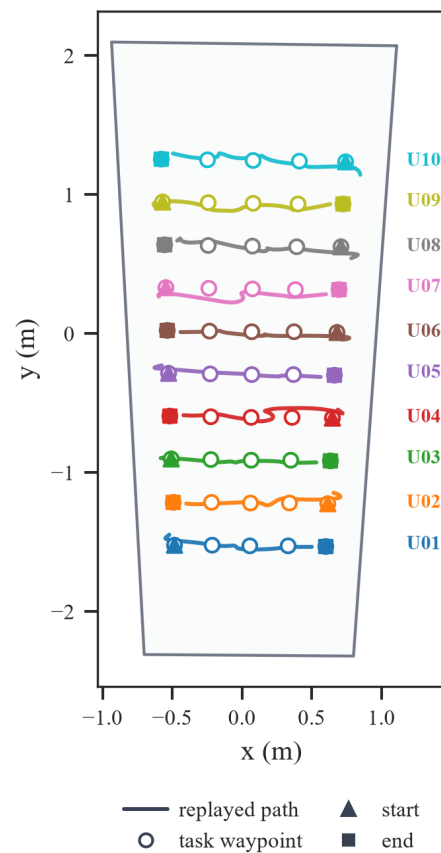


**Figure 8.** Indoor hardware platform for the real-UAV replay experiments, including the motion-capture sensing layer, relay computer, host computer, and UAV execution layer.

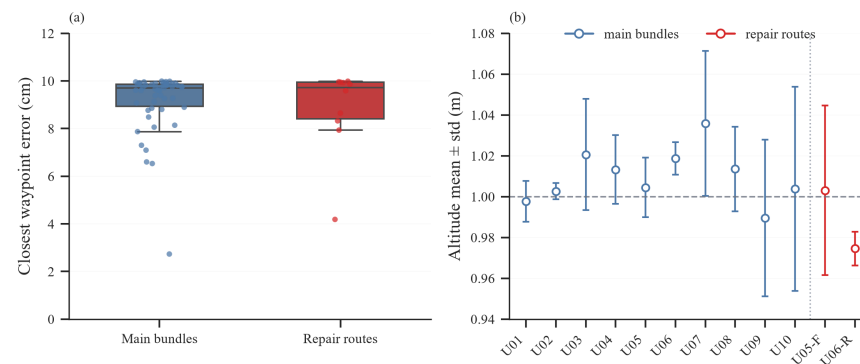
Figure 9 reports the physical replay of the nominal route sequences. The ten virtual route sequences obtained from the 10-UAV/50-task planning case are executed sequentially on the real platform, producing 10 route replays and 50 task waypoints in total. The replayed trajectories remain close to the designated waypoints throughout the tested runs. All 50 task waypoints satisfy the 0.25 m threshold, yielding a task completion rate of 100%. The mean and maximum closest waypoint errors are 9.16 cm and 9.99 cm, respectively, and the mean replay duration is 9.72 s per route. These results indicate that the routes generated by the allocation layer can be transferred to waypoint-level execution without evident loss of geometric consistency in the tested indoor setup, but they do not validate online multi-UAV reassignment or inter-UAV deconfliction.

Figure 10 quantifies the replay accuracy and height stability observed in the physical tests. In Figure 10a, the closest horizontal waypoint errors for both nominal routes and repaired routes remain far below the 0.25 m acceptance criterion, and most observations cluster near 0.10 m. In Figure 10b, the mean altitude and its standard deviation remain centered around 1 m for all replay runs, indicating that the low-level flight controller preserves stable vertical motion while the high-level planner outputs are issued as waypoint

commands. Taken together, these results show that the observed replay deviation is not dominated by systematic drift or loss of altitude in the tested setup.



**Figure 9.** Real-UAV repeated physical replay of the main virtual multi-UAV route sequences.



**Figure 10.** Closest horizontal waypoint error and altitude statistics from real-UAV replay logs: (a) distributions of closest waypoint error for the main bundles and repair routes; (b) mean altitude with standard deviation for each replay. In (b), the horizontal dashed line marks the 1.0 m reference altitude, and the vertical dotted line separates the main-bundle and repair-route replays.

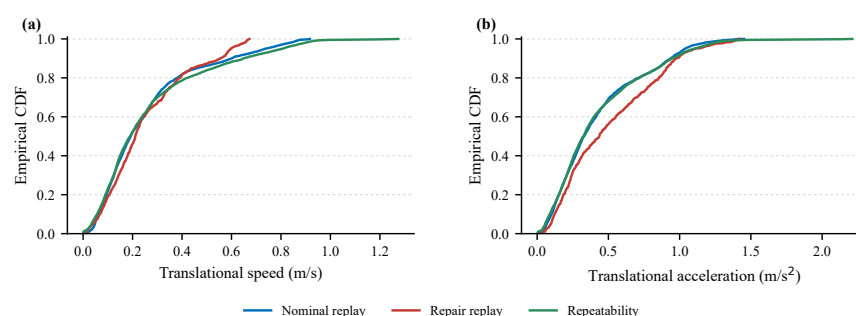
To examine the translational motion recorded during replay, all 15 motion-capture logs were resampled on a uniform time grid using the median sampling interval of each run. Position signals were smoothed with a third-order Savitzky–Golay filter over 21 samples, corresponding to approximately 0.70 s at the median sampling rate of 30.00 Hz, before first and second derivatives were calculated. Half of the filter window was excluded at both record boundaries. This procedure produced 6990 valid derivative samples. It describes observed translational kinematics and does not reconstruct unlogged control

inputs or vehicle attitude dynamics. Table 6 summarizes the resulting translational-speed and acceleration statistics.

**Table 6.** Observed translational kinematics derived from the motion-capture replay logs.

| Protocol       | Logs | Speed (m/s) |       |       | Acceleration (m/s <sup>2</sup> ) |       |       |
|----------------|------|-------------|-------|-------|----------------------------------|-------|-------|
|                |      | Mean        | 95th  | Max.  | Mean                             | 95th  | Max.  |
| Nominal replay | 10   | 0.258       | 0.737 | 0.917 | 0.425                            | 1.038 | 1.452 |
| Repair replay  | 2    | 0.254       | 0.597 | 0.673 | 0.508                            | 1.110 | 1.414 |
| Repeatability  | 3    | 0.270       | 0.807 | 1.274 | 0.430                            | 1.104 | 2.212 |

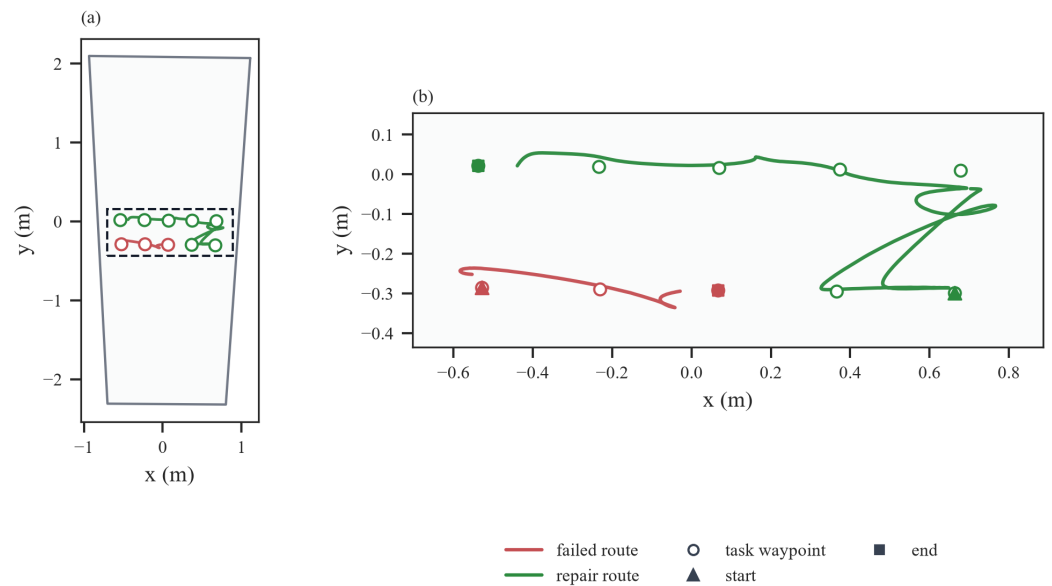
Across all replay logs, the pooled mean, 95th-percentile, and maximum translational speeds are 0.264, 0.765, and 1.274 m/s, respectively; the corresponding accelerations are 0.434, 1.071, and 2.212 m/s<sup>2</sup>. Figure 11 shows that the nominal, repair, and repeatability protocols occupy comparable observed ranges. Yaw is constant within every available log, with a maximum within-run range of 0 rad, so no yaw-rate inference is made. Rotor thrust, attitude rate, actuator saturation, and control-input signals were not logged. These measurements add translational-kinematics evidence to the waypoint-error analysis, but they do not prove full nonlinear dynamic feasibility.



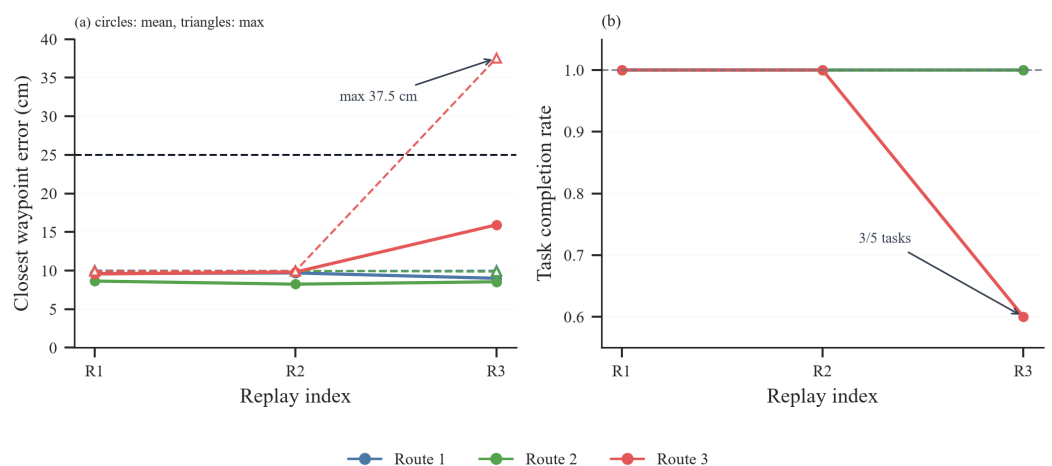
**Figure 11.** Empirical cumulative distributions derived from the 15 motion-capture replay logs: (a) translational speed; (b) translational acceleration.

Figure 12 illustrates the physical replay of failure-triggered route repair. This test mirrors the reallocation logic of the UAV-failure stress scenario, but it is executed offline because the current platform does not support online onboard redistribution during flight. Figure 12a marks the repair region within the full workspace, whereas Figure 12b compares the interrupted route segment with the repaired route delivered after redistribution. Two repair-related routes are executed, covering 10 repair-stage task waypoints in total. All 10 task waypoints satisfy the 0.25 m threshold, and the mean closest waypoint error is 8.38 cm. The evidence supports waypoint-level replay of the repaired task sequences in the tested indoor setting, but not online multi-UAV redistribution or nonlinear dynamic feasibility.

Figure 13 evaluates repeatability by replaying three representative routes three times each. For Route 1 and Route 2, both the mean closest waypoint error and the task completion rate remain nearly unchanged across repeated executions. Route 3 exhibits a clear degradation in the third replay, where one task waypoint is missed and the maximum closest waypoint error rises to 37.54 cm. Across the nine repeated flights, the overall task completion rate is 95.56%, and the mean closest waypoint error is 9.89 cm. The repeatability results therefore support stable execution in most runs while also exposing the operating limit of the current open-loop replay procedure under accumulated disturbance and run-to-run variation.



**Figure 12.** Real-UAV replay of failure-triggered repaired routes: (a) full workspace with the repair region indicated; (b) enlarged repair region comparing the failed and repaired routes.



**Figure 13.** Repeatability of real-UAV physical replay across three representative routes: (a) mean closest waypoint error (circles and solid lines) and maximum error (triangles and colored dashed lines), with the horizontal dashed line marking the 25 cm acceptance threshold; (b) task completion rate. In (b), the Route 1 (blue) and Route 2 (green) curves coincide at 100% completion, so the blue curve is not separately visible.

## 5. Conclusions

This article developed ET-PBBA for event-local multi-UAV task allocation under task releases, static obstacles, deadlines, limited energy, and UAV failures. The method combines  $A^*$ -based static-obstacle path cost, predicted energy, and secondary load–energy pressure in a one-pass feasible-pair assignment. Controlled tests show that event-time activation increases completion rate by 0.046 in the main scenario and 0.065 under failure stress relative to 10 s periodic allocation. Adding the physical-cost block to reward-only ranking improves completion by 0.239 and 0.245, whereas the balance increment is not significant in the ordinary scenarios. Under the common comparison protocol, ET-PBBA reduces average completion time by 5.2% and 4.1% and total energy by 4.3% and 3.0% relative to auction and CBBA, respectively. The completion-rate differences are not significant, and Hungarian remains the stronger centralized reference. Under failure stress, ET-PBBA attains a completion rate of 0.790 with 9.73 deadline violations. Single-UAV replay completes 50 nominal

waypoints with 100% completion and 9.16 cm mean error, while nine repeatability runs achieve 95.56% completion and 9.89 cm mean error. Across 15 logs, the 95th-percentile translational speed and acceleration are 0.765 m/s and 1.071 m/s<sup>2</sup>. These measurements support waypoint replay only. The current model excludes lossy communication, nonlinear flight feasibility, and synchronous multi-UAV execution. Future work will address these limitations together with inter-UAV deconfliction.

**Author Contributions:** Conceptualization, Y.C. and Y.L.; methodology, Y.C.; software, Y.C.; validation, Y.C. and X.Z.; formal analysis, Y.C. and Z.W.; investigation, Y.C.; resources, Z.W. and Y.L.; data curation, Y.C.; writing—original draft preparation, Y.C.; writing—review and editing, Y.L., X.Z. and Z.W.; visualization, Y.C.; supervision, Z.W.; project administration, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available from the corresponding author upon reasonable request. The simulation scripts, raw simulation records, and real-UAV replay logs used to generate the reported tables and figures are retained by the authors.

**Conflicts of Interest:** Author Yujia Li was employed by the company Commercial Aircraft Corporation of China. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

## References

- Gerkey, B.P.; Mataric, M.J. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **2004**, *23*, 939–954. [\[CrossRef\]](#)
- Korsah, G.A.; Stentz, A.; Dias, M.B. A comprehensive taxonomy for multi-robot task allocation. *Int. J. Robot. Res.* **2013**, *32*, 1495–1512. [\[CrossRef\]](#)
- Nunes, E.; Manner, M.; Mitiche, H.; Gini, M. A taxonomy for task allocation problems with temporal and ordering constraints. *Robot. Auton. Syst.* **2017**, *90*, 55–70. [\[CrossRef\]](#)
- Skaltsis, G.; Shin, H.-S.; Tsourdos, A. A review of task allocation methods for UAVs. *J. Intell. Robot. Syst.* **2023**, *109*, 76. [\[CrossRef\]](#)
- Alqefari, S.; Menai, M.E.B. Multi-UAV task assignment in dynamic environments: Current trends and future directions. *Drones* **2025**, *9*, 75. [\[CrossRef\]](#)
- Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.-H.; Debbah, M. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2334–2360. [\[CrossRef\]](#)
- Zeng, Y.; Zhang, R. Energy-efficient UAV communication with trajectory optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3747–3760. [\[CrossRef\]](#)
- Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [\[CrossRef\]](#)
- Bertsekas, D.P. The auction algorithm: A distributed relaxation method for the assignment problem. *Ann. Oper. Res.* **1988**, *14*, 105–123. [\[CrossRef\]](#)
- Dias, M.B.; Zlot, R.; Kalra, N.; Stentz, A. Market-based multirobot coordination: A survey and analysis. *Proc. IEEE* **2006**, *94*, 1257–1270. [\[CrossRef\]](#)
- Choi, H.-L.; Brunet, L.; How, J.P. Consensus-based decentralized auctions for robust task allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–926. [\[CrossRef\]](#)
- Capitan, J.; Spaan, M.T.J.; Merino, L.; Ollero, A. Decentralized multi-robot cooperation with auctioned POMDPs. *Int. J. Robot. Res.* **2013**, *32*, 650–671. [\[CrossRef\]](#)
- Qamar, R.A.; Sarfraz, M.; Rahman, A.; Ghauri, S.A. Multi-criterion multi-UAV task allocation under dynamic conditions. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101734. [\[CrossRef\]](#)
- Song, Y.; Ma, Z.; Chen, N.; Zou, S.; Srigrarom, S. Comparative analysis of centralized and distributed multi-UAV task allocation algorithms: A unified evaluation framework. *Drones* **2025**, *9*, 530. [\[CrossRef\]](#)
- Huang, H.; Jiang, Z.; Yan, T.; Bai, Y. Dynamic task allocation for heterogeneous multi-UAVs in uncertain environments based on 4DI-GWO algorithm. *Drones* **2024**, *8*, 236. [\[CrossRef\]](#)

16. Zhang, J.; Chen, Y.; Yang, Q.; Lu, Y.; Shi, G.; Wang, S.; Hu, J. Dynamic task allocation of multiple UAVs based on improved A-QCDPSO. *Electronics* **2022**, *11*, 1028. [\[CrossRef\]](#)
17. Wu, J.; Zhang, J.; Sun, Y.; Li, X.; Gao, L.; Han, G. Multi-UAV collaborative dynamic task allocation method based on ISOM and attention mechanism. *IEEE Trans. Veh. Technol.* **2024**, *73*, 6225–6235. [\[CrossRef\]](#)
18. Liu, D.; Dou, L.; Zhang, R.; Zhang, X.; Zong, Q. Multi-agent reinforcement learning-based coordinated dynamic task allocation for heterogeneous UAVs. *IEEE Trans. Veh. Technol.* **2023**, *72*, 4372–4383. [\[CrossRef\]](#)
19. Yu, J.; Zhang, Y.; Sun, C. Multi-UAV dynamic task assignment based on event-triggered graph reinforcement learning under weak communication. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 19106–19121. [\[CrossRef\]](#)
20. Ghauri, S.A.; Sarfraz, M.; Qamar, R.A.; Sohail, M.F.; Khan, S.A. A review of multi-UAV task allocation algorithms for a search and rescue scenario. *J. Sens. Actuator Netw.* **2024**, *13*, 47. [\[CrossRef\]](#)
21. Yan, S.; Feng, J.; Pan, F. A distributed task allocation method for multi-UAV systems in communication-constrained environments. *Drones* **2024**, *8*, 342. [\[CrossRef\]](#)
22. Dong, N.; Liu, S.; Mai, X. Communication-efficient heterogeneous multi-UAV task allocation based on clustering. *Comput. Commun.* **2025**, *229*, 107986. [\[CrossRef\]](#)
23. Yang, M.; Zhang, A.; Bi, W.; Wang, Y. A resource-constrained distributed task allocation method based on a two-stage coalition formation methodology for multi-UAVs. *J. Supercomput.* **2022**, *78*, 10025–10062. [\[CrossRef\]](#)
24. Zhen, Z.; Wen, L.; Wang, B.; Hu, Z.; Zhang, D. Improved contract network protocol algorithm based cooperative target allocation of heterogeneous UAV swarm. *Aerosp. Sci. Technol.* **2021**, *119*, 107054. [\[CrossRef\]](#)
25. Wang, G.; Lv, X.; Yan, X. A two-stage distributed task assignment algorithm based on contract net protocol for multi-UAV cooperative reconnaissance task reassignment in dynamic environments. *Sensors* **2023**, *23*, 7980. [\[CrossRef\]](#)
26. Yang, M.; Bi, W.; Zhang, A.; Gao, F. A distributed task reassignment method in dynamic environment for multi-UAV system. *Appl. Intell.* **2022**, *52*, 1582–1601. [\[CrossRef\]](#)
27. Neville, A.; Chernova, S.; Ravichandar, H. D-ITAGS: A dynamic interleaved approach to resilient task allocation, scheduling, and motion planning. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1037–1044. [\[CrossRef\]](#)
28. Hu, C.; Li, Y.; Qu, G. Integrated dynamic task allocation via event-triggered for tracking ground moving targets by UAVs in urban. *Robot. Auton. Syst.* **2025**, *193*, 105061. [\[CrossRef\]](#)
29. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [\[CrossRef\]](#)
30. Koenig, S.; Likhachev, M. Fast replanning for navigation in unknown terrain. *IEEE Trans. Robot.* **2005**, *21*, 354–363. [\[CrossRef\]](#)
31. Sharon, G.; Stern, R.; Felner, A.; Sturtevant, N.R. Conflict-based search for optimal multi-agent pathfinding. *Artif. Intell.* **2015**, *219*, 40–66. [\[CrossRef\]](#)
32. Xu, G.; Kang, X.; Yang, H.; Wu, Y.; Liu, W.; Cao, J.; Liu, Y. Distributed multi-vehicle task assignment and motion planning in dense environments. *IEEE Trans. Autom. Sci. Eng.* **2024**, *21*, 5837–5852. [\[CrossRef\]](#)
33. Lu, Z.; Zhou, T.; Mou, S. Real-time multi-robot mission planning in cluttered environment. *Robotics* **2024**, *13*, 40. [\[CrossRef\]](#)
34. Feng, S.; Zeng, L.; Liu, J.; Yang, Y.; Song, W. Multi-UAVs collaborative path planning in the cramped environment. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 529–538. [\[CrossRef\]](#)
35. Wang, K.; Gong, Y.; Li, C.; Liu, P. Joint optimization of task allocation, power optimization and trajectory planning in multi-UAV system. *Phys. Commun.* **2026**, *74*, 102926. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.