I.J. Wireless and Microwave Technologies, **2026, 5, 37-47**
Published Online on October 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2026.05.03

Learning-Augmented Deterministic Task Allocation and Routing for Heterogeneous Multi-UAV Systems

Mykola Nikolaiev*
Department of computer engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, 03056, Ukraine
E-mail: nickolay.dev@gmail.com
ORCID iD: https://orcid.org/0009-0003-0787-6726
*Corresponding Author

Mykhailo Novotarskyi
Department of computer engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, 03056, Ukraine
E-mail: novotar@gmail.com
ORCID iD: https://orcid.org/0000-0002-5653-8518

Received: April 3, 2026; Revised: June 26, 2026; Accepted: July 19, 2026; Published: October 8, 2026

Abstract: This paper presents a learning-augmented deterministic method for heterogeneous multi-UAV task allocation and routing in static, fully observed environments. UAV-specific asymmetric path-length matrices represent the output of an offline path planner, allowing online planning to focus on assignment and route construction. The deterministic core evaluates insertions, checks energy feasibility, prioritizes tasks with few feasible UAVs, repairs blocked states, and improves completed routes. A shallow multilayer perceptron ranks feasible candidates by predicting future regret, defined as the difference between the makespan after deterministic completion and the immediate post-insertion makespan. The objective minimizes makespan first and uses aggregate route length only when makespan values are equal within the prescribed numerical tolerance. The model was trained on 12 synthetic instances and evaluated on 20 separate instances, each containing 20 tasks and four heterogeneous UAVs under uniform, clustered, and mixed layouts. It was compared with two greedy insertion heuristics, the corresponding deterministic ablation (CT-D), and a linear-regret variant. The proposed method achieved a mean makespan of 106.24 versus 107.46 for CT-D, a descriptive reduction of approximately 1.1%. Compared with CT-D, aggregate route length increased by approximately 1.5%, while mean runtime increased from 0.059 to 0.233 s. Performance improved on clustered and mixed layouts but declined slightly on uniform layouts.

Index Terms: Heterogeneous UAVs, multi-UAV routing, task allocation, optimization, learning-augmented optimization, future-regret prediction

1. Introduction

Multi-UAV mission planning requires joint task allocation and route construction. An assignment should not be judged only by its immediate insertion cost because it also affects route structure, energy use, workload balance, and the feasibility of remaining tasks. This coupling is repeatedly emphasized in recent multi-UAV mission-planning and task-allocation reviews, where auction-based, optimization-based, game-theoretic, learning-based, and metaheuristic methods are classified as major solution families [1–4]. For the present study, these broad classifications must be complemented by a comparison of how individual algorithms handle route feasibility, vehicle heterogeneity, and non-myopic decision-making.

The problem becomes especially difficult in heterogeneous fleets. In the setting studied here, UAVs differ in speed, energy budget, energy consumption per unit distance, and directed path lengths between nodes. Therefore, a locally inexpensive assignment may consume the limited energy budget of a UAV that has few feasible alternatives for later tasks. Classical routing heuristics address part of this issue through insertion and regret principles [5–7], while

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

heterogeneous UAV scheduling and distributed allocation methods address heterogeneity in task assignment [8–10]. Nevertheless, these works do not fully match the setting considered here: a centralized deterministic planner with UAV-specific asymmetric path-length matrices, deterministic energy-feasibility checks, and candidate ranking based on downstream effects on mission makespan.

A second group of related algorithms is represented by decentralized auction and consensus-based approaches. The consensus-based bundle framework in [11] and its extensions for task coupling, replanning, time windows, clustering, and dynamic heterogeneous allocation [12–14,17,19] are highly relevant when distributed execution and communication efficiency are central requirements. These methods are scalable and robust in multi-agent settings, but their decision logic is usually based on local bids, bundle construction, or consensus updates. As a result, they do not directly address the specific question studied in this paper: how a centralized deterministic planner should choose among several already feasible insertion candidates when their immediate costs are similar but their future route-level consequences differ.

Integrated task-assignment and path-planning approaches provide another important point of comparison. The work in [15] jointly optimizes task assignment and UAV paths using genetic algorithms and motion constraints. Later methods incorporate emergency uncertainty, trajectory optimization, deadlock handling, communication constraints, and richer three-dimensional motion models [16,18,23]. These approaches are appropriate when allocation and trajectory design must be solved jointly online. The present paper uses a different decomposition: UAV-specific ordered-pair path lengths are assumed to be available before online planning, as motivated by [22], and the online algorithm uses these data for deterministic assignment and route construction.

Learning-enhanced task allocation has also advanced significantly. The method in [20] is particularly close because it combines learning with a classical optimizer: a Transformer-based NeuroSelect heuristic guides a discrete particle swarm optimization process. The two-stage method in [21] similarly combines graph modeling with a learning-inspired immune algorithm. These studies show that learning can improve multi-UAV allocation, but they generally embed learning inside a broader stochastic optimizer or policy-guided search procedure. In contrast, the present work deliberately restricts learning to a narrower role. The neural model does not generate routes, does not replace feasibility checks, and does not control a population-based optimizer. It predicts only the future regret of candidates that have passed deterministic feasibility screening.

Accordingly, this paper presents a hybrid planner that combines deterministic feasible insertion, repair, and local improvement with supervised ranking of feasible candidates. The learned component estimates downstream makespan effects, while route construction and feasibility control remain deterministic. The specific contribution is an empirical evaluation of whether deterministic-completion labels can improve candidate ranking without replacing the planner's core decision logic.

2. Related Works

Surveys [1–4] classify the principal families of multi-UAV task-allocation methods, but they provide limited comparison of the mechanisms most relevant to this study: deterministic feasibility, UAV-specific route data, and non-myopic insertion decisions.

The routing foundation of the proposed method is closest to constructive insertion and regret-based vehicle-routing heuristics. The work in [5] demonstrated that insertion heuristics are effective for vehicle routing and scheduling with time-window constraints because they build routes incrementally while maintaining feasibility. Reference [6] extended this logic through a generalized regret measure, where the next insertion is selected not only by immediate cost but also by the penalty of losing a good insertion opportunity. Reference [7] showed that route construction can be strengthened through adaptive large-neighborhood repair and improvement operators. These works justify the use of insertion, regret, and route-improvement principles. Nevertheless, their formulations were developed for classical routing settings and do not directly model heterogeneous UAV fleets, UAV-specific asymmetric path-length matrices, or energy-feasible task insertion under mission makespan minimization. The present paper adapts this constructive logic to multi-UAV routing by evaluating feasible insertions deterministically and using future-regret prediction to estimate downstream makespan effects.

Heterogeneous task allocation has been addressed more directly in [8-10]. The scheduling formulation in [8] treats heterogeneous UAV task allocation as a combinatorial optimization problem and highlights that vehicle capability differences strongly affect feasible schedules. The decentralized surveillance method in [9] is important because it explicitly considers critical tasks, showing that not all tasks should be treated equally during allocation. Reference [10] proposes a heuristic distributed allocation method for multivehicle multitask problems and applies it to search and rescue. These works move beyond general taxonomies by addressing heterogeneity, task criticality, and distributed feasibility. However, they mainly focus on task assignment or distributed decision-making rather than path-aware route insertion using precomputed UAV-specific ordered-pair path lengths. The present method differs by defining task criticality through the number of feasible assignees remaining at each construction step and then ranking only the corresponding feasible insertion candidates.

Auction-based and consensus-based allocation methods form the most established decentralized line of work. The method in [11] introduced consensus-based decentralized auctions and the consensus-based bundle algorithm, enabling conflict-free task allocation through local bidding and consensus. Reference [12] extended this direction to heterogeneous multi-UAV systems with task coupling constraints. Reference [13] incorporated local replanning for time-sensitive and dynamic environments, while [14] developed distributed allocation under time-window constraints. More recent variants improve scalability and communication efficiency. Reference [17] uses clustering to decompose heterogeneous multi-UAV allocation and reduce communication overhead, while [19] introduces a two-level clustered consensus-based bundle algorithm for dynamic heterogeneous multi-task allocation. These methods are important alternatives for distributed or communication-constrained missions. Their limitation relative to the present paper is not poor performance, but a different design goal: they optimize allocation through bids, bundles, clustering, and consensus, whereas the proposed planner focuses on deterministic centralized route construction with exact energy checks and learned prediction of future regret.

Integrated task-assignment and path-planning methods provide a closer comparison in terms of coupling allocation with motion. The genetic-algorithm approach in [15] jointly optimizes task assignment and path planning for cooperating UAVs and incorporates motion constraints through a Dubins vehicle model. Reference [16] addresses emergency rescue planning with uncertain requirements by combining robust optimization and intelligent path-planning algorithms. Reference [18] integrates task assignment and trajectory planning using a three-dimensional Dubins-path representation and deadlock handling. Reference [23] considers task allocation and trajectory optimization for multi-UAV cargo systems under cellular-connected constraints. These works model physical motion, uncertainty, connectivity, or trajectory feasibility more directly than the present paper. However, this realism usually increases online optimization complexity and can make the solution framework scenario-specific. The present study instead assumes that UAV-specific path-length data are available offline, as motivated by [22], and uses these data during online assignment and route construction. This preserves path awareness without performing trajectory optimization at every insertion decision.

Metaheuristic and learning-enhanced methods are also highly relevant. The method in [20] proposes a learning improvement heuristic for multi-UAV task allocation in which a Transformer-based NeuroSelect mechanism guides a discrete particle swarm optimization process through reinforcement learning. This is one of the closest related methods because it also combines learning with a classical optimizer. However, learning in [20] is embedded in the evolutionary search process, whereas the present paper uses a shallow supervised model only to score feasible deterministic insertion candidates. Reference [21] proposes a two-stage framework that combines graph theory with a learning-inspired immune algorithm, improving task-allocation search through structural modeling and adaptive optimization. Compared with [20] and [21], the proposed method is more transparent at inference: candidate generation, feasibility checking, route updating, repair, and local improvement remain deterministic.

Reference [22] motivates the use of offline UAV-specific route data. In the present study, however, [22] serves only as a model for the data representation: the benchmark matrices are generated synthetically, and the allocation results are evaluated independently of the performance of the path-planning method.

Recent benchmark-oriented work also clarifies why the proposed method should be positioned carefully. Reference [24] argues for unified evaluation of centralized and distributed multi-UAV task-allocation algorithms because different algorithmic families have different strengths under different assumptions. This observation is directly relevant here. The proposed method is not intended to replace distributed CBBA variants in communication-constrained missions, nor to replace full trajectory-optimization frameworks in highly dynamic environments. Its target setting is narrower but well defined: centralized static planning with known tasks, deterministic path-length matrices, heterogeneous UAV parameters, exact energy feasibility, and a makespan-oriented objective.

The literature therefore leaves a narrower gap. Classical insertion heuristics are fast and interpretable but rank candidates mainly by immediate effects, whereas learning-enhanced methods commonly embed learning within broader stochastic search. This study evaluates whether a lightweight supervised model can improve candidate ranking while deterministic logic retains route construction and feasibility control.

3. Proposed Approach

The proposed planner addresses heterogeneous multi-UAV task allocation and routing in a static, fully observed environment. UAV-specific route data for all relevant ordered node pairs are assumed to be available before online planning. Online computation then performs task assignment and route construction. A supervised model estimates the downstream effect of each feasible insertion, while candidate generation, feasibility checking, repair, and route updating remain deterministic. The learned score approximates the outcome of one-step CT-D completion. Figure 1 shows this division of responsibilities.

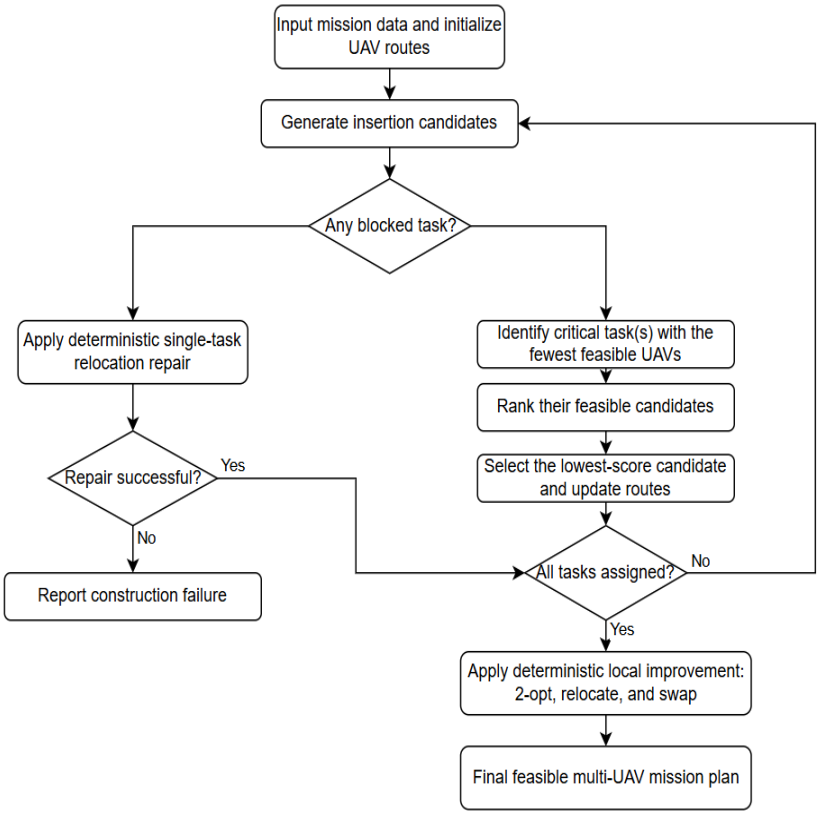


Fig. 1. Overview of the proposed deterministic planner

Let $\mathcal{U} = 1, \dots, m$ denote the UAV fleet and $\mathcal{T} = 1, \dots, n$ denote the task set. UAV $u \in \mathcal{U}$ starts at s_u and terminates at g_u; the common-depot case is obtained by setting $s_u = g_u = d_0$ for every UAV. For each ordered node pair (i, j), the offline route-data layer provides a nonnegative path length c_{ij}^u for UAV u. The resulting matrix $C^u = [c_{ij}^u]$ may be asymmetric, and all route evaluations use its directed entries without assuming $c_{ij}^u = c_{ji}^u$. UAV u has speed v_u, energy budget B_u, and linear energy coefficient ρ_u. For a route $r_u = (n_0^u, \dots, n_{h_u}^u)$, where $n_0^u = s_u$ and $n_{h_u}^u = g_u$, its route length, completion time, and energy consumption are

$$L_u(r_u) = \sum_{l=0}^{h_u-1} c_{n_l^u n_{l+1}^u}^u, \qquad T_u(r_u) = \frac{L_u(r_u)}{v_u}, \qquad E_u(r_u) = \rho_u L_u(r_u) \tag{1}$$

A solution $R = \{r_u\}_{u \in \mathcal{U}}$ is feasible when every task is visited exactly once and $E_u(r_u) \le B_u$ for every UAV. Let Ω denote the set of feasible solutions. The primary objective is the mission makespan $C_{max}(R) = \max_{u \in \mathcal{U}} T_u(r_u)$ and the secondary objective is the aggregate route length $D(R) = \sum_{u \in \mathcal{U}} L_u(r_u)$. The optimization criterion is explicitly lexicographic:

$$\operatorname{lex\,min}_{R \in \Omega} \big(C_{max}(R), D(R) \big) \tag{2}$$

The comparison is implemented lexicographically with a fixed numerical tolerance. A solution is preferred when its makespan is lower by more than this tolerance; aggregate route length is used only when the makespan difference falls within the tolerance. This avoids an arbitrary weighted sum and reflects the operational priority of fleet completion time.

The planner constructs the routes iteratively. At iteration t, the partial state is $S^t = (R^t, W^t)$, where R^t is the current route set and $W^t \subseteq \mathcal{T}$ is the set of unassigned tasks. Initially, $r_u^0 = (s_u, g_u)$ for every UAV and $W^0 = \mathcal{T}$. For each task $i \in W^t$, each UAV u, and each pair of consecutive route nodes n_l^u and n_{l+1}^u, the exact insertion-cost change is

$$\Delta L_{uil}^t = c_{n_l^u i}^u + c_{i n_{l+1}^u}^u - c_{n_l^u n_{l+1}^u}^u \tag{3}$$

The corresponding time and energy increments are $\Delta T_{uil}^t = \Delta L_{uil}^t / v_u$ and $\Delta E_{uil}^t = \rho_u \Delta L_{uil}^t$. An insertion is feasible only if $E_u(r_u^t) + \Delta E_{uil}^t \le B_u$. For each task–UAV pair, only the feasible position with the smallest ΔL_{uil}^t is retained; equal-cost positions are resolved by the smaller route index. Let $\mathcal{F}_i^t$ denote the set of UAVs that can feasibly accept task i at iteration t. The planner first considers tasks with the smallest value of $|\mathcal{F}_i^t|$. All tasks tied for the minimum are considered. Here, "critical" denotes limited assignment flexibility, not mission priority. This critical-task rule prioritizes tasks with limited assignment flexibility and is intended to reduce the risk that they become impossible to assign at a later construction stage.

Each retained candidate is represented by eight deterministic and interpretable features. Four features describe its immediate effect: the insertion-time increase ΔT_{ui}^t; the makespan immediately after insertion; the post-insertion energy-utilization ratio $\frac{E_u(r_u^t)+\rho_u \Delta L_{ui}^t}{B_u}$; and the current route load, defined as $\frac{T_u(r_u^t)}{C_{max}(R^t)+\epsilon}$ where $\epsilon > 0$ prevents division by zero. The remaining four features describe assignment structure. Task isolation is the minimum averaged symmetric travel time between task i and another unassigned task, where the pairwise value is defined as $\bar{\tau}_{ij} = (2m)^{-1} \sum_{u \in \mathcal{U}} \frac{\left(c_{ij}^w + c_{ji}^w\right)}{v_w}$. When only one task remains unassigned, isolation is measured relative to the nearest UAV start or terminal node using the same bidirectional convention. The assignment margin is the difference between the candidate's immediate post-insertion makespan and the smallest immediate post-insertion makespan obtainable for the same task among all UAVs in $\mathcal{F}_i^t$. The flexibility ratio of UAV u is the fraction of currently unassigned tasks that it can still accept feasibly. Finally, relative insertion fit measures how the task fits between its selected predecessor p and successor q, and is calculated as $\frac{c_{pq}^u}{c_{pi}^u + c_{iq}^u + \epsilon}$. Larger values indicate that visiting the task introduces a relatively small detour with respect to the replaced route connection. All features are computed from the current partial state and the precomputed matrices. Before training and inference, every feature is standardized using the mean and standard deviation calculated from the training set only.

The learned component is a multilayer perceptron with eight inputs, one hidden layer of 32 rectified linear units, and one scalar linear output. Its target is called future regret throughout this paper. Here, future regret denotes the downstream change in final makespan after the immediate insertion effect has been accounted for. It is distinct from cumulative regret in online learning.

Training examples are generated offline using deterministic rollouts. Partial states are produced by running the critical-task deterministic planner (CT-D) on the training instances. CT-D uses the same feasible-candidate generation, critical-task prioritization, repair procedure, and final local search as the proposed method, but it ranks candidates only by their immediate post-insertion makespan and does not use a learned prediction. At every sampled partial state, each feasible candidate $a = (u, i, \mathrm{l})$ is temporarily applied and the resulting state $S^t \oplus a$ is completed using the deterministic CT-D completion operator. This operator repeatedly applies the CT-D selection rule until all remaining tasks have been assigned and then applies the same deterministic local-improvement procedure used during evaluation. The future-regret label is

$$\Gamma_a^t = C_{max}\left(\mathcal{D}_{\text{CT-D}}(S^t \oplus a)\right) - C_{max}(S^t \oplus a) \tag{4}$$

The first term in (4) is the final makespan obtained by completing the candidate state with CT-D, whereas the second term is the exact makespan immediately after the candidate insertion. The label therefore isolates the downstream effect that cannot be observed from the immediate post-insertion makespan alone. If deterministic completion remains infeasible after repair, the candidate is assigned a fixed infeasibility value larger than every feasible future-regret label in the training data; this value is fixed before model fitting and is not adjusted using test instances.

The training data are divided by complete mission instance rather than by individual candidate record, thereby preventing closely related partial states from the same instance from appearing in both training and validation sets. The model is fitted by minimizing mean squared error between predicted and observed future regret. The feature-scaling parameters, network parameters, calibration coefficient, training seed, optimizer settings, stopping rule, and validation split are fixed before test evaluation and retained unchanged for all test instances.

During online planning, the model predicts $\widehat{\Gamma_a^t}$ for each feasible candidate associated with the currently most critical tasks. The primary candidate score is

$$\widehat{Q_a^t} = C_{max}(S^t \oplus a) + \beta \widehat{\Gamma_a^t} \tag{5}$$

$\beta \ge 0$ is selected using training or validation data only. The candidate with the smallest $\widehat{Q_a^t}$ is selected. Consistent with (2), equal primary scores are resolved first by the lower post-insertion aggregate route length, followed by the smaller immediate insertion cost, lower post-insertion energy utilization, task index, UAV index, and insertion position. These fixed tie-breaking rules ensure deterministic inference.

Equation (5) also provides a direct interpretation of the learned component. When $\beta = 1$ and the predicted future regret equals the label in (4), $\widehat{Q_a^t}$ is exactly the makespan produced by a one-step CT-D rollout. The model therefore

approximates the repeated deterministic completion of every feasible candidate state. Theoretically, if the predictor's error is bounded such that $\left|\widehat{\Gamma}_a^t - \Gamma_a^t\right| \leq \delta$ for all valid candidates, the true rollout makespan of the selected candidate is at most 2δ greater than that of the rollout-best candidate within the same critical candidate set. This is a local ranking property rather than a global optimality guarantee, because CT-D is itself heuristic and only candidates associated with the most critical tasks are evaluated.

Setting $\beta = 0$ removes the learned term and recovers the CT-D baseline without changing any other planning component. The linear-regret baseline uses the same features, labels, deterministic rollout procedure, and online ranking rule, but replaces the multilayer perceptron with a linear regression model. These ablations isolate the effect of learned look-ahead from the effects of critical-task prioritization, feasibility checking, repair, and local improvement.

A blocked state occurs when an unassigned task has no feasible UAV. In this case, the planner invokes a deterministic single-task relocation repair. For the blocked task, the procedure enumerates an already assigned task whose removal from a donor route would make the blocked insertion feasible, together with every feasible recipient UAV and insertion position for the removed task. All affected directed route lengths are recomputed, and combinations violating an energy budget are discarded. The feasible repair with the smallest resulting makespan is selected, with aggregate route length and fixed task, UAV, and position indices used as successive tie-breakers. If no such relocation exists, the planner reports construction failure rather than returning an infeasible solution.

After all tasks have been assigned, deterministic local search evaluates intra-route 2-opt moves and inter-route relocate and swap moves. Because the path-length matrices may be asymmetric, the lengths of the affected routes are recalculated using the directed matrix entries rather than symmetric 2-opt formulas. A move is accepted only when all affected routes remain energy-feasible and the resulting solution is lexicographically better according to (2). At each iteration, the best improving move is selected, and the procedure terminates when no admissible move reduces the makespan or, at equal makespan, reduces aggregate route length.

All route lengths, energy checks, repairs, and local-search decisions are therefore computed deterministically. The neural network supplies only an approximate downstream ranking signal for candidates that are already feasible. For the fixed network architecture, each prediction adds constant per-candidate work and does not change the asymptotic order of the insertion procedure. Its practical runtime effect is evaluated in Section 4. Finally, feasibility is evaluated only with respect to the supplied path-length matrices and the linear energy model in (1); aerodynamic effects, communication constraints, collision avoidance, and battery dynamics are outside the present model.

4. Experiments and Results

The proposed method was evaluated on a fixed synthetic benchmark designed to reflect the assumptions of the planning framework. The test set comprised 20 instances, each containing $n = 20$ tasks and $m = 4$ heterogeneous UAVs. Three spatial task distributions were considered: uniform, clustered, and mixed. The mixed family combined dense task groups with isolated tasks and was intended to represent instances in which early assignment decisions may substantially affect the feasibility and quality of later insertions. For each spatial family, moderate and high levels of UAV heterogeneity were examined by varying vehicle speed and effective energy range.

For every instance, UAV-specific asymmetric path-length matrices were generated for all ordered node pairs to emulate route data that could be produced by an offline path planner [22]. All methods used the same task locations, UAV parameters, and path-length matrices. Consequently, differences in solution quality arose from the task-allocation and route-construction logic rather than from differences in the input route data.

The learned models were trained offline on 12 additional synthetic instances that were separate from the 20 test instances. The proposed method used the nonlinear future-regret predictor described in Section 3, whereas the linear-regret variant used the same features and rollout-generated labels but replaced the multilayer perceptron with a linear regression model.

Five methods were compared. Greedy distance insertion selected the feasible candidate with the smallest increase in route length. Greedy makespan insertion selected the candidate with the smallest immediate post-insertion makespan. CT-D used the same critical-task prioritization, deterministic feasibility checks, repair procedure, and final local improvement as the proposed method but excluded the learned future-regret term. The linear-regret variant retained the complete deterministic framework and learned look-ahead mechanism but used a linear predictor. The proposed method used the nonlinear predictor with one hidden layer.

\

The primary metric was final mission makespan C_{max}. Aggregate route length D was the secondary metric under the lexicographic objective, and runtime measured computational overhead. Lower values are preferable for all three metrics, although aggregate route length must be interpreted jointly with makespan because the planner prioritizes fleet completion time.

Table 1. Descriptive aggregate results

Method	Mean makespan C_{max}	Mean aggregate route length D	Mean runtime (s)
Greedy distance	140.58±4.37	301.05±5.34	0.076±0.011
Greedy makespan	126.97±3.93	344.91±3.26	0.074±0.013
CT-D	107.46±2.14	366.73±3.53	0.059±0.009
Linear-regret	106.42±2.22	382.69±3.14	0.199±0.031
Proposed method	106.24±1.57	372.15±2.16	0.233±0.120

The proposed method achieved the lowest mean makespan, with a value of 106.24, compared with 107.46 for CT-D. This corresponds to an absolute reduction of 1.22 and a relative reduction of approximately 1.1%. The improvement is therefore modest. Table 1 reports variation across test instances as standard deviations. However, no confidence intervals or paired statistical tests were computed; therefore, the 1.1% difference should be interpreted as descriptive rather than statistically confirmed.

The linear-regret variant achieved a mean makespan of 106.42, only 0.18 greater than that of the proposed nonlinear model. The present results therefore do not demonstrate a substantial advantage of nonlinear regression over a linear predictor. Instead, they suggest that the principal benefit may arise from introducing a future-regret signal into the deterministic candidate-ranking rule. Establishing whether the nonlinear model offers a reliable additional benefit would require paired instance-level analysis and repeated training runs with different random seeds.

Both learned variants achieved substantially lower mean makespan than the two greedy reference methods. Relative to greedy distance insertion, the proposed method reduced mean makespan by approximately 24.4%, while the reduction relative to greedy makespan insertion was approximately 16.3%. These comparisons show that the complete framework performed better than simple local insertion rules on the tested benchmark, but they do not isolate the contribution of any single component. CT-D and the linear-regret variant provide the more informative internal comparisons. No external state-of-the-art method was evaluated.

Figure 2 provides a family-wise comparison between CT-D and the proposed method. To make the relative effect comparable across spatial families with different absolute makespan levels, the CT-D mean was normalized to 100 within each family. The normalized value of the proposed method was calculated relative to the corresponding CT-D family mean. Values below 100 therefore indicate an improvement over CT-D, whereas values above 100 indicate a deterioration.

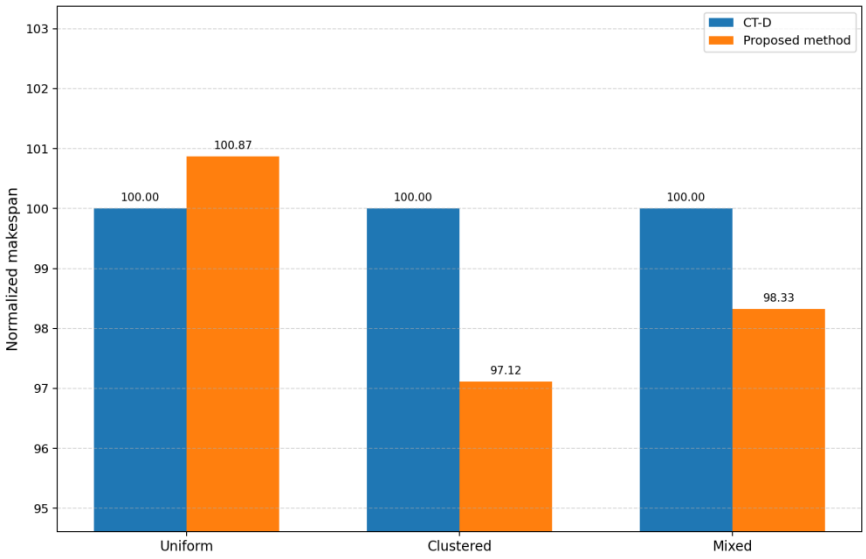


Fig. 2. Normalized family-wise mean makespan

On uniform instances, the normalized makespan of the proposed method was 100.87, indicating that it performed approximately 0.87% worse than CT-D. On clustered instances, the normalized value was 97.12, corresponding to a reduction of approximately 2.88%. On mixed instances, the value was 98.33, corresponding to a reduction of approximately 1.67%. In absolute terms, the mean makespan on clustered instances decreased from 97.75 for CT-D to 94.93 for the proposed method, while on mixed instances it decreased from 110.08 to 108.24.

The family-wise results show that the benefit of the learned ranking term was not uniform across all spatial distributions. The largest improvement occurred on clustered instances, followed by mixed instances, whereas no improvement was obtained on the uniform family. This pattern is consistent with the motivation for future-regret prediction. In clustered and mixed environments, several locally similar insertion choices may have different

consequences for the remaining tasks, particularly when isolated tasks or unevenly distributed clusters restrict the set of suitable UAVs. In such cases, a learned estimate of downstream effects may provide useful information that is not captured by immediate post-insertion makespan alone. In more regular uniform layouts, the deterministic insertion rule may already represent much of the relevant routing structure, leaving less opportunity for learned look-ahead to improve the solution.

The aggregate route-length results illustrate the trade-off created by the makespan-first objective. Greedy distance insertion achieved the lowest mean aggregate route length, 301.05, but also the largest mean makespan, indicating that local distance minimization did not balance UAV completion times effectively. The proposed method increased mean aggregate route length from 366.73 for CT-D to 372.15, or approximately 1.5%, while reducing mean makespan by approximately 1.1%.

The linear-regret model obtained the highest mean aggregate route length among the three critical-task-based methods, with $D = 382.69$, while its mean makespan remained slightly higher than that of the proposed method. At the aggregate level, the nonlinear model therefore provided a more favorable combination of makespan and aggregate route length than the linear-regret variant. However, the difference is small and cannot be assessed reliably without paired distributions.

The runtime results show a measurable computational cost associated with learned candidate ranking. CT-D required 0.059 s per instance on average, whereas the proposed method required 0.233 s. The proposed method was therefore approximately 3.9 times slower than its deterministic ablation, although the absolute increase was only 0.174 s. The linear-regret variant required 0.199 s, which was closer to the nonlinear method than to CT-D. The similar runtimes of the linear and nonlinear variants suggest that feature extraction and repeated prediction calls may account for much of the overhead. Profiling would be required to confirm this interpretation.

Overall, the proposed method produced the lowest mean makespan and retained sub-second mean runtime in the reported implementation. However, the improvement over CT-D was small, the nonlinear model was only marginally better than linear regression, and performance deteriorated slightly on uniform instances. The results are therefore preliminary and do not establish a statistically reliable advantage.

5. Conclusion

This study presented a learning-augmented deterministic framework for heterogeneous multi-UAV task allocation and routing in static, fully observed mission environments. The framework separates offline route-data generation from online task assignment and route construction. During online planning, deterministic components perform feasible insertion evaluation, energy-budget checking, critical-task prioritization, blocked-state repair, and local route improvement. Learning is restricted to ranking feasible insertion candidates according to their predicted future regret, defined as the downstream makespan effect that remains after the immediate consequence of an insertion has been considered. This restricted role preserves deterministic feasibility and route construction while introducing a limited form of look-ahead into the candidate-selection process.

The method was evaluated on 20 synthetic test instances, each containing 20 tasks and four heterogeneous UAVs under uniform, clustered, and mixed spatial distributions. Among the five evaluated approaches, the proposed method achieved the lowest mean makespan. Its mean makespan was 106.24, compared with 107.46 for the deterministic CT-D baseline, corresponding to an absolute reduction of 1.22 and a relative reduction of approximately 1.1%. This improvement was accompanied by an increase in mean aggregate route length from 366.73 to 372.15, or approximately 1.5%, and an increase in mean runtime from 0.059 s to 0.233 s. The additional route length is compatible with the makespan-first lexicographic objective because balancing UAV completion times may require longer routes overall. However, aggregate results alone do not establish whether this trade-off was favorable on every test instance.

The comparison with the linear-regret variant provides additional insight into the role of the learned component. The linear model achieved a mean makespan of 106.42, only 0.18 higher than that of the nonlinear model. The current evidence therefore does not establish a substantial advantage of the multilayer perceptron over linear regression. Instead, the results suggest that much of the observed improvement may be associated with introducing a downstream ranking signal into the deterministic planner, rather than with the nonlinear architecture itself. This finding is relevant from an engineering perspective because the linear model may offer a simpler and more interpretable alternative if future paired tests show no reliable difference between the two predictors.

The family-wise results show that the effect of learned ranking depends on the spatial structure of the tasks. Relative to CT-D, the proposed method reduced mean makespan by approximately 2.9% on clustered instances and 1.7% on mixed instances, whereas it increased normalized mean makespan by approximately 0.9% on uniform instances. This pattern is consistent with the motivation for future-regret prediction. Clustered and mixed layouts can create uneven assignment opportunities, isolated tasks, and stronger dependencies between current insertions and the feasibility of later decisions. Under these conditions, candidates with similar immediate post-insertion makespans may produce different downstream outcomes, giving the learned ranking signal an opportunity to improve the deterministic choice.

The slight deterioration on uniform instances requires particular attention. In more regular layouts, immediate post-insertion makespan and route-length change may already provide sufficiently informative candidate rankings, while structural features such as task isolation and assignment flexibility may vary only slightly between candidates. The learned term may therefore add limited information and occasionally change an otherwise appropriate deterministic decision. The small training set may also reduce the model's ability to distinguish between near-equivalent candidates. These explanations are plausible but remain hypotheses because the present study did not examine candidate-level prediction errors, ranking accuracy, or individual route changes. A reliable explanation will require a larger set of uniform instances and direct analysis of the decisions on which the proposed method and CT-D disagree.

The empirical findings should therefore be interpreted as preliminary. The test benchmark is small, and although Table 1 reports variability across instances, the evaluation does not include confidence intervals, paired statistical tests, or repeated training runs with different random seeds. The baseline set also consists primarily of simple greedy methods and internal ablations, without comparison against an external exact solver, a competitive metaheuristic, or another learning-augmented routing method. In addition, the experiments use synthetic precomputed route data, a linear energy model, fixed tasks, and complete mission information. These assumptions support controlled evaluation of the proposed ranking mechanism but limit conclusions about operational UAV missions.

Future work should first strengthen the empirical evaluation. A larger paired benchmark is needed to assess performance across different task counts, fleet sizes, heterogeneity levels, and degrees of path-length asymmetry. Per-instance results, confidence intervals, effect sizes, win–tie–loss counts, and paired statistical tests should be reported. Repeated training runs are also necessary to distinguish variation caused by mission instances from variation caused by model fitting. Comparisons with stronger external methods would clarify whether the observed improvement remains meaningful beyond the selected deterministic baseline and simple greedy reference methods.

A second priority is to investigate when learned ranking should influence the deterministic planner. Candidate-level prediction errors and ranking disagreements should be analyzed separately for uniform, clustered, and mixed layouts. Based on this analysis, a confidence-aware fallback could retain the CT-D decision when the learned predictions are uncertain or when candidate scores are nearly equal. This would preserve the benefits observed on structured instances while reducing the risk of unnecessary changes on regular layouts. Such a mechanism should be evaluated before introducing more complex model architectures or larger feature sets.

Further extensions should address dynamic and physically realistic mission settings. A rolling-horizon version could update assignments when tasks, UAV availability, or route estimates change during execution. The route and energy models should also be validated using higher-fidelity simulation or flight-derived data that account for factors such as wind, payload, altitude changes, battery behavior, and vehicle-specific motion constraints. These extensions would help determine whether the observed behavior under synthetic route data transfers to operational environments.

Within the stated scope, the results indicate that a narrowly targeted learned ranking component can be integrated into a deterministic multi-UAV planner without replacing its feasibility logic or route-construction mechanisms. The proposed method achieved the lowest mean makespan among the evaluated approaches and was most effective on clustered and mixed task layouts. Nevertheless, the improvement over CT-D was modest and has not yet been statistically confirmed. The main contribution should therefore be understood as a transparent proof of concept for learning-guided candidate ranking within deterministic multi-UAV routing, rather than as conclusive evidence of superior performance across general mission settings.

All the Declarations and Statements

Author Contributions Statement
Both authors made substantial contributions.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement
The authors declare no conflict of interest

Funding Declaration
This research received no external funding.

Data Availability Statement
N/A

Ethical Declaration
N/A

Acknowledgments
N/A

Declaration of generative AI in scholarly writing
Generative AI tools were used to assist in improving the clarity, grammar, and readability of the manuscript. The authors take full responsibility for the content of the work.

Abbreviations
The following abbreviations are used in this manuscript:

UAV - Unmanned Aerial Vehicle
CT-D - Critical-Task Deterministic

Appendix A/B/C…, with appendix title
N/A

References

[1] J. Song, K. Zhao, and Y. Liu, "Survey on Mission Planning of Multiple Unmanned Aerial Vehicles," Aerospace, vol. 10, no. 3, p. 208, 2023. https://doi.org/10.3390/aerospace10030208
[2] G. M. Skaltsis, H.-S. Shin, and A. Tsourdos, "A Review of Task Allocation Methods for UAVs," Journal of Intelligent & Robotic Systems, vol. 109, p. 76, 2023. https://doi.org/10.1007/s10846-023-02011-0
[3] S. A. Ghauri, M. Sarfraz, R. A. Qamar, M. F. Sohail, and S. A. Khan, "A Review of Multi-UAV Task Allocation Algorithms for a Search and Rescue Scenario," Journal of Sensor and Actuator Networks, vol. 13, no. 5, p. 47, 2024. https://doi.org/10.3390/jsan13050047
[4] M. Rinaldi, S. Wang, R. S. Geronel, and S. Primatesta, "Application of Task Allocation Algorithms in Multi-UAV Intelligent Transportation Systems: A Critical Review," Big Data and Cognitive Computing, vol. 8, no. 12, p. 177, 2024. https://doi.org/10.3390/bdcc8120177
[5] M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," Operations Research, vol. 35, no. 2, pp. 254–265, 1987. https://doi.org/10.1287/opre.35.2.254
[6] J.-Y. Potvin and J.-M. Rousseau, "A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows," European Journal of Operational Research, vol. 66, no. 3, pp. 331–340, 1993. https://doi.org/10.1016/0377-2217(93)90221-8
[7] S. Røpke and D. Pisinger, "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows," Transportation Science, vol. 40, no. 4, pp. 455–472, 2006. https://doi.org/10.1287/trsc.1050.0135
[8] K. Li, "Heuristic Task Scheduling on Heterogeneous UAVs: A Combinatorial Optimization Approach," Journal of Systems Architecture, vol. 140, p. 102895, 2023. https://doi.org/10.1016/j.sysarc.2023.102895
[9] G. Binetti, D. Naso, and B. Turchiano, "Decentralized Task Allocation for Surveillance Systems with Critical Tasks," Robotics and Autonomous Systems, vol. 61, no. 12, pp. 1653–1664, 2013. https://doi.org/10.1016/j.robot.2013.06.007
[10] W. Zhao, Q. Meng, and P. W. H. Chung, "A Heuristic Distributed Task Allocation Method for Multivehicle Multitask Problems and Its Application to Search and Rescue Scenario," IEEE Transactions on Cybernetics, vol. 46, no. 4, pp. 902–915, 2016. https://doi.org/10.1109/TCYB.2015.2418052
[11] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-Based Decentralized Auctions for Robust Task Allocation," IEEE Transactions on Robotics, vol. 25, no. 4, pp. 912–926, 2009. https://doi.org/10.1109/TRO.2009.2022423
[12] F. Ye, J. Chen, Q. Sun, Y. Tian, and T. Jiang, "Decentralized Task Allocation for Heterogeneous Multi-UAV System with Task Coupling Constraints," The Journal of Supercomputing, vol. 77, pp. 111–132, 2021. https://doi.org/10.1007/s11227-020-03264-4
[13] J. Chen, X. Qing, F. Ye, K. Xiao, K. You, and Q. Sun, "Consensus-Based Bundle Algorithm with Local Replanning for Heterogeneous Multi-UAV System in the Time-Sensitive and Dynamic Environment," The Journal of Supercomputing, vol. 78, pp. 1712–1740, 2022. https://doi.org/10.1007/s11227-021-03940-z
[14] W. Cui, R. Li, Y. Feng, and Y. Yang, "Distributed Task Allocation for a Multi-UAV System with Time Window Constraints," Drones, vol. 6, no. 9, p. 226, 2022. https://doi.org/10.3390/drones6090226
[15] E. Edison and T. Shima, "Integrated Task Assignment and Path Optimization for Cooperating Uninhabited Aerial Vehicles Using Genetic Algorithms," Computers & Operations Research, vol. 38, no. 1, pp. 340–356, 2011. https://doi.org/10.1016/j.cor.2010.06.001
[16] M. Cao, J. Peng, L. Yang, and Y. Yang, "Multi-UAV Task Allocation and Path Planning in Emergency Rescue Scenarios with Uncertain Requirements," Journal of Industrial and Management Optimization, vol. 21, no. 10, pp. 6107–6133, 2025. https://doi.org/10.3934/jimo.2025124
[17] N. Dong, S. Liu, and X. Mai, "Communication-Efficient Heterogeneous Multi-UAV Task Allocation Based on Clustering," Computer Communications, vol. 229, p. 107986, 2025. https://doi.org/10.1016/j.comcom.2024.107986
[18] W. Wu, L. Zhang, J. Le, and Z. Lu, "Integrated Method for Multi-UAV Task Assignment and Trajectory Planning with Deadlock Based on Three-Dimensional Dubins Path," Scientific Reports, vol. 15, p. 24152, 2025. https://doi.org/10.1038/s41598-025-09753-x

[19] Y. Wang, C. Wang, and S. Ren, "A Two-Level Clustered Consensus-Based Bundle Algorithm for Dynamic Heterogeneous Multi-UAV Multi-Task Allocation," Sensors, vol. 25, no. 21, p. 6738, 2025. https://doi.org/10.3390/s25216738
[20] B. Fan, Y. Bo, and X. Wu, "Learning Improvement Heuristics for Multi-Unmanned Aerial Vehicle Task Allocation," Drones, vol. 8, no. 11, p. 636, 2024. https://doi.org/10.3390/drones8110636
[21] S. Zhang, C. Hu, D. Zhao, K. Yang, Z. Xu, and M. Li, "A Two-Stage Multi-UAV Task Allocation Approach Based on Graph Theory and a Learning-Inspired Immune Algorithm," Drones, vol. 9, no. 9, p. 599, 2025. https://doi.org/10.3390/drones9090599
[22] M. Nikolaiev, M. Novotarskyi, and A. Volokyta, "A Novel Hybrid Differential Evolution and Enhanced Whale Optimization Algorithm for UAV Path Planning," International Journal of Information Technology and Computer Science, vol. 17, no. 6, pp. 109–126, 2025. https://doi.org/10.5815/ijitcs.2025.06.06
[23] B. Zhang, K. Huang, Y. Chen, and D. Yang, "Task Allocation and Trajectory Optimization for Multi-UAV Cargo Systems with Cellular-Connected Constraints," IET Communications, vol. 19, no. 1, 2025. https://doi.org/10.1049/cmu2.70106
[24] Y. Song, Z. Ma, N. Chen, S. Zhou, and S. Srigrarom, "Comparative Analysis of Centralized and Distributed Multi-UAV Task Allocation Algorithms: A Unified Evaluation Framework," Drones, vol. 9, no. 8, p. 530, 2025. https://doi.org/10.3390/drones9080530

Authors' Profiles



Mykola Nikolaiev PhD student, Department of computer engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, 03056, Ukraine. Areas of scientific interests: Distributed systems, optimization methods, algorithms, machine learning.



Mykhailo Novotarskyi Doctor of Sciences, Professor, Department of computer engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, 03056, Ukraine. Areas of scientific interests: Mathematical modeling of complex physical processes, formal description of mathematical and simulation models, parallel locally asynchronous numerical methods, machine learning and data mining, reinforcement learning methods.

How to cite this paper

Mykola Nikolaiev, Mykhailo Novotarskyi, "Learning-Augmented Deterministic Task Allocation and Routing for Heterogeneous Multi-UAV Systems", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.5, pp. 37-47, 2026. DOI:10.5815/ijwmt.2026.05.03