Article

Integrated Consensus Framework for Task Assignment and Path Planning of a Degraded UAV Fleet

Waleed T. Barnawi [1], Gillian M. Nicholls [2,*], Charles D. McAllister [2,†] and Alisha Y. Ortiz [3]

[1] United States Office of Naval Research, 875 N. Randolph St., Arlington, VA 22203, USA; wbarnawi@gmail.com (W.T.B.)

[2] Department of Marketing, Southeast Missouri State University, One University Plaza MS 5875, Cape Girardeau, MO 63701, USA; cdmcallister@fhsu.edu (C.D.M.)

[3] Department of Engineering & Technology, Southeast Missouri State University, One University Plaza MS 6825, Cape Girardeau, MO 63701, USA; aortiz@semo.edu (A.Y.O.)

* Corresponding author. E-mail: gnicholls@semo.edu (G.M.N.)

† Present: Robbins College of Business & Entrepreneurship, Fort Hays State University, 600 Park Street, McCartney Hall 302, Hays, KS 67601, USA.

Received: 28 July 2025; Accepted: 19 September 2025; Available online: 28 September 2025

ABSTRACT: Unmanned aerial vehicle (UAV) systems can fail during civil and military operations. This presents a significant challenge for human teleoperators (remote pilots) in determining task reallocation after member loss within the fleet. To alleviate the high cognitive load on teleoperators in critical situations, a decentralized strategy was developed to resolve the combined task assignment and vehicle routing problems. This Integrated Consensus Framework (ICF) not only solves the combined problem but also adds a unique ability to identify the loss of a vehicle and dynamically reroute agents to abandoned tasks to achieve a satisfactory solution. ICF is a two-tiered approach that combines a novel algorithm, the Caravan Auction (CarA) algorithm, with a path-planning strategy to identify when UAVs are lost and reallocate orphaned tasks. The CarA Algorithm consists of three phases: auction, consensus, and validation phases. An experiment using Monte Carlo simulations was conducted to determine the performance of ICF. Teleoperators assigned to complete multiple tasks with UAVs in dangerous environments can allow the proposed system to perform task assignments and reallocation while offering only supervisory control as needed. The results indicate this novel approach provides comparable performance to existing strategies, doing so with the addition of randomized UAV loss.

Keywords: Unmanned aircraft systems; Multi-vehicle intelligent systems; Coordinated control

1. Introduction

Unmanned aerial vehicles (UAVs) are being heavily utilized for a growing number of applications in both the civil and military arenas. Examples include domestic traffic monitoring [1], surveying and assessing land and buildings [2], border patrols [3], security surveillance [4], agriculture [5,6], parcel delivery [7–9], wildfire monitoring [10], entertainment [11], and military operations [12].

The vehicle routing problem (VRP) is a generalization of the Traveling Salesman Problem, which seeks to determine how a single agent can efficiently traverse multiple points only once. In the VRP, a fleet of identical vehicles with finite capacity must deliver goods from a central depot to N_c dispersed customers [13]. Over the years, researchers have studied this problem and its variations [14]. One well-known variation is the dynamic Vehicle Routing Problem, which presents real-world situations in which vehicles are forced to adjust in transit in response to changes in the problem state, such as schedule changes, new demands, or lost vehicles. This problem has been studied extensively, leading to numerous solutions [15–18]. Further extensions to this problem have been explored for application in UAVs [19–24]. Hino and Tsuchiya investigated aerodynamically efficient path planning for unmanned aerial vehicle (UAV) fleets [25]. Coordination between unmanned aerial and ground vehicles has been explored [26], as well as the ability of a UAV fleet to conduct a cooperative ground mission [27]. Taran and Homaeinezhad examined adaptive consensus to

https://doi.org/10.70322/dav.2025.10016

control the actions of multiple UAVs [28]. Chen, Du, Zhang, & Wei modeled a coverage path planning approach for choosing paths of a mixed fleet of varying UAVs types by clustering the regions to be serviced [29].

The assignment problem (AP) is a fundamental combinatorial optimization problem that involves finding an optimal matching in a weighted bipartite graph. In these problems, elements from one set are matched with elements from another set. This is most generally observed when certain types of agents are matched with tasks, where the goal is usually to maximize a predefined objective function. Many studies have investigated this area as a centralized problem [30], a decentralized problem [31,32], dynamic behavior in the environment [33,34], and consensus among agents [19,35]. A number of these studies have proposed solutions using auction algorithms. Brunet [35] developed the Consensus-Based Bundle Algorithm (CBBA), a decentralized way to have a fleet of agents "bid" on individual tasks assembled into a bundle for that agent, with assignments made based on the calculated score for each task. This was an extension of an approach first proposed by Alighanbari [36]. Later work by Choi, Brunet, & How [37] demonstrated the excellent performance of CBBA compared to prior auction algorithms for task assignment. CBBA has been adapted by Zhang, Feng, Shi, Jiang, Chowdhury, & Ling [38] for use by UAV swarms with new constraints such as task deadlines, different task categories, emergence of new tasks, and UAV capacity constraints. Chen, Qing, Ye, Xiao, You, & Sun [39] examined extending CBBA to include local replanning as CBBA-LR. The goal was to allow tasks to be reassigned when a new, time-sensitive task emerged, as might be expected in emergencies such as search and rescue functions. Jung [40] considered the possibility that a UAV might be lost due to system failure or using up its fuel after task assignment. Li, Ren, Shi, Lin, & Qin [41] explored a different threat that has become more pressing—the vulnerability of agents to cyber attacks such as Denial of Service (DoS) efforts by a hostile force to disrupt the communications that networked multi-agent systems need for effective operation.

Auction algorithms offer an intuitive approach to solving classical assignment problems. In this approach, agents bid on tasks. Bids are generally resolved by calculating the net value of a task's rewards. Improvements to naive auction algorithms have been proposed [42]. Further advances in auction algorithms have led to solutions to variations in assignment problems [32]. Often, for UAVs, AP and VRP do not exist independently [19,35,43] but are tightly coupled. Li, Yue, Shi, Lin, & Qin [44] utilized neural networks to aid multi-agent systems dealing with uncertainty in target location in a dynamic environment of mobile targets and variations in speed. The potential for using UAVs organized in swarms has necessitated even more complex coordination [45–50]. Generational advances in communication technology with the advent of 5G mobile networks have supported greater capacity for connected devices and correspondingly greater communication integration for UAVs [51]. The anticipated 6G wireless technology is expected to continue expanding the utility of multiple UAV systems [52,53].

1.1. Motivation

As applications become more complex, the systems that perform them must also become correspondingly complex. Given their increased complexity, these systems are prone to failure. Furthermore, specific applications of UAVs can be dangerous and jeopardize the safety of these systems. Examples of such tasks include fire surveillance and military applications. The U.S. RQ-170 incident, in which the government of Iran allegedly took control to crash a UAV, provided clear evidence that these systems were not impervious [54]. Therefore, it is important to mitigate these potential threats for risky applications.

1.2. Contribution

This study extends previous work by other researchers [35,37] and proposes a methodology to determine when an agent may be lost, and its task must be reassigned. In previous approaches, auction algorithms have been combined with consensus algorithms to perform task allocation. An extension was provided to address the path planning using Dijkstra's algorithm. This study introduces a novel two-tiered strategy called the Integrated Consensus Framework (ICF) [55]. It comprises a new three-phased methodology termed the Caravan Auction (CarA) Algorithm for the AP and Dijkstra's Algorithm for the VRP. The multiple assignment CarA technique extends the Consensus-Based Auction Algorithm (CBAA) developed by Brunet [35] by providing awareness for lost resources, prioritizing remaining tasks, and adopting abandoned tasks. The differences from prior research include the following:

1. Task validation for executed assignments—Nanjanath and Gini [31] incorporated validation of task execution, but not the precedence of the tasks.

2. Identification and fleet-wide notification of lost UAVs—Jung [40] considered lost resources but had a centralized task assignment process. Brunet [35] and Rabbath [43] utilized a decentralized task assignment, but they did not allow for lost resources.
3. Reallocation of tasks to available agents and re-routing to the new tasks—Brunet [35] and Choi, Brunet, and How [37] developed efficient task allocation assignments, but did not extend to reallocation as agents dropped out of communication with the fleet (either temporarily or permanently) or were lost.
4. Better performance than benchmark studies for some of the factors of interest including the optimality gap, the number of convergence steps, and the mean and standard deviation of execution times.

Section 2 provides mathematical formulations for task assignment and vehicle routing problems. Further mathematical descriptions of the decentralized auctions and consensus algorithms are presented. In Section 3, the ICF is decomposed along with algorithmic descriptions. Section 4 provides a brief description of the Monte Carlo simulation. In Section 5, the ICF is compared with two benchmark algorithms. Section 6 summarizes our findings regarding ICF.

2. Problem Formulation

The combined task assignment and vehicle routing problem is decomposed for mathematical formulation.

2.1. Task Assignment Formulation

The task assignment problem is a combinatorial optimization problem generally described as the minimization (or maximization) of a global objective while adhering to a conflict-free assignment of N_m tasks to N_u agents. For each agent i assigned to task j, the associated net value score s_{ij}. A binary variable describes the assignment with x_{ij} equal to 1 if agent i is assigned to task j and 0 otherwise. In this scenario, the objective is to maximize the overall score of the agents. When presented as a linear programming problem, the AP is formulated as:

$$max \sum_{i=1}^{N_u}\sum_{j=1}^{N_m} s_{ij}x_{ij} \tag{1}$$

$$\text{subject to } \forall i = \{1, \dots, N_u\}\text{: } \textstyle\sum_{j=1}^{N_m} x_{ij} \le 1$$

$$\forall j = \{1, \dots, N_m\}\text{: } \sum_{i=1}^{N_u} x_{ij} \le 1$$

$$\forall i = \{1, \dots, N_u\},\ \forall j = \{1, \dots, N_m\}\text{: } x_{ij} \in \{0{,}1\}$$

2.2. Auction Algorithms

Auction algorithms are combinatorial optimization algorithms established as a methodology to address both AP and VRP using linear or nonlinear costs. A sales auction is conducted by determining the best price (cost) for a set of products (tasks) proposed by a set of bidders (agents). These auctions are regulated in centralized and decentralized manners.

Auction algorithms seek to determine an economic equilibrium by assigning all tasks to the highest bid from each agent. Each agent acts in a greedy manner. Consider that task j has a score s_{ij} which is the difference between the reward a_{ij} for assigning task j to agent i and price p_j of task j.

$$s_{ij} = a_{ij} - p_j \tag{2}$$

The value of price updates reflects the current bid as the assignment procedure progresses. In the naive auction algorithm, shown in Equation (2), updates in rounds are initialized with a randomly assigned set of tasks and agents. Using the sales auction analogy, while an agent must pay price p_j for task j, it believes that it has acquired the maximal value for task j if:

$$j^* = \underset{j}{argmax}\{a_{ij} - p_j\} \tag{3}$$

If this condition holds and task j^* is unassigned, agent *i* is assigned task j^* and is satisfied. This simple methodology holds true until agents run into a situation in which two agents can continually swap tasks, resulting in a churning behavior. To resolve this potential issue, the price of task j^* is increased, such that the task price is now:

$$p_{j^*} + \gamma_i \tag{4}$$

$$\text{where } \gamma_i = v_i - w_i \tag{5}$$

v_i is the best score for agent *i*:

$$v_i = \max_j \{a_{ij} - p_j\} \tag{6}$$

and w_i is the second-best score for agent *i*:

$$w_i = \max_{j \neq j^*} \{a_{ij} - p_j\} \tag{7}$$

The bidding increment γ_i sets the price for the assigned task such that agent *i* views no difference between task j^* and the second-highest score task. The bidder's preferred task is now considered less attractive to potential competitors. One flaw in this approach is that bidding increments of zero occur when more than one task offers a maximal value to an agent. Bertsekas [29] resolved this problem by introducing a perturbation mechanism that ensures that each bid for a task raises its price by a minimum positive increment, ε, thereby resolving an issue with a zero-bidding increment. This method resembles the naïve auction algorithm, except that the bidding increment γ_i is:

$$\gamma_i = v_i - w_i + \varepsilon \tag{8}$$

In the generalized auction algorithm case, the auctioneer is considered to have full knowledge of bidders' prices and states. Yet many situations warrant a methodology when communication to the fleet of agents is not continual, and price updates are not immediate upon assignment. These cases call for decentralized auction algorithms, where the score for each agent/task pairing is calculated as:

$$s_{ij} = a_{ij} - p_{ij} \tag{9}$$

In this instance, price p_{ij} is now associated with agent *i* instead of being a generalized value. Similarly, the value of score s_{ij} is also localized. In this approach, the auctioneer is shifted from a centralized server to agents acting as auctioneers. The auctioneer determines winner i^* based on the highest bid:

$$i^* = \underset{i}{argmax}\{s_{ij}\} \tag{10}$$

This process is iterated until all tasks are appropriately assigned.

2.3. Consensus Algorithms

Maintaining consistent situational awareness in decentralized systems is an important and challenging task. Proper coordination involves having each agent be aware of the other agents' states and the surrounding environment. In the physical layer, data synchronization provides the necessary situational awareness to perform cooperative and coordinated actions among teams. Data synchronization occurs at either the input or the output of a coordinated algorithm. Several issues complicate the data synchronization efforts, including unreliable communication links and range limitations. Consensus algorithms provide a method for decentralized systems to converge on important information such as target locations and vehicle states.

Beard and Stepanyan [56] defined the conditions for data synchronization or what will become the consensus algorithm using a set of agents $U = \{u_1, \dots, u_n\}$ such that the data are synchronized if the information variables *r* for each agent u_i and u_j demonstrate:

$$\|r_i(t) - r_j(t)\| = 0 \tag{11}$$

where r_i is the information variable for agent *i*. The set of agents *U* is globally asymptotically synchronized if, for any $r_i(0)$, $\underset{t\to\infty}{\|r_i(t) - r_j(t)\|} \to 0$ for all pairs of agents.

The information variables are assumed to be continuously updated using the updated law over time, as follows:

$$\dot{r}_i = f_i\big(r_1, r_2, \dots, r_N, G(t)\big) \tag{12}$$

where $G(t)$ is an adjacency matrix representing the communication topology at time t. Adjacency matrices are a convenient way of representing directed graphs. The adjacency matrix $G(t)$ has a binary representation in a matrix format such that:

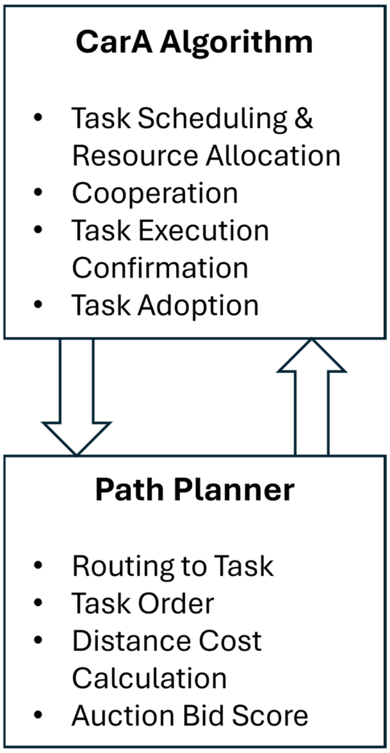
$$G_{ij} = \begin{cases} 1, \text{if there is an edge from } u_i \text{ to } u_j \\ 0, \text{otherwise} \end{cases} \tag{13}$$

The matrix is also considered to be piecewise constant at time *t* but is dynamic in nature with random changes in its elements over its duration.

3. Integrated Consensus Framework (ICF)

ICF is a two-tiered hierarchical framework designed to resolve combined task and vehicle routing problems. This framework extends the existing paradigm of hierarchical frameworks by providing a unique, real-time solution for UAVs (and other agents) that lose resources during a mission. A simple functional diagram of the framework is shown in Figure 1 illustrating the relationship between the upper and lower levels. The top layer introduces the new CarA Algorithm, which provides high-level functionality including task scheduling and resource allocation.

The algorithm ensures cooperation among the fleet of agents and the notification of task execution through broadcast messages. If an agent is lost, the algorithm also ensures that orphaned tasks are reassigned. The path-planning layer conducts a low-level computational effort, which includes the calculation of costs from an agent to tasks and the optimal paths to a task. Cost is used to determine the bid score for an agent/task assignment.



CarA Algorithm

- Task Scheduling & Resource Allocation
- Cooperation
- Task Execution Confirmation
- Task Adoption

Path Planner

- Routing to Task
- Task Order
- Distance Cost Calculation
- Auction Bid Score

Figure 1. Integrated Consensus Framework Architecture.

3.1. Caravan Auction (CarA) Algorithm

The CarA Algorithm comprises three phases, as shown in Figure 2. The algorithm is a single-assignment approach that combines aspects of auction and consensus algorithms with a novel validation strategy developed to ensure task completion. This figure provides an illustrative description of the interactions among the three phases.

The first element is the auction phase, in which each agent places a bid on the available tasks within its capability to complete. After the agent wins one of its bids, it enters the consensus phase. This ensures that the task is assigned to the most qualified agent. In a dynamic environment, where agents in motion enter and leave the communication range, this phase must be iterated repeatedly, and agents must compare their bids and simultaneously move into the validation phase. If an agent is outbid, it surrenders the task and restarts the auction.

The validation phase is an original contribution that monitors the task completion time limits and task completion in general. Each agent in the consensus phase, which compares bids on tasks, is also in the validation phase and monitors task status updates, including its own. When an assignment is completed, an agent in this phase notifies its peers of the task completion. Each agent in contact notes the completion. If an agent fails to complete its task within the presumed time limit, which includes a buffer for lost connectivity, the agent is presumed to be lost and its task is released. This newly orphaned task is added to the set of unassigned tasks, and unassigned agents compete for it. If the originally assigned agent returns to connectivity before time expires, the process continues. Alternatively, if it returns to connectivity after its time limit, it must compete for new tasks.

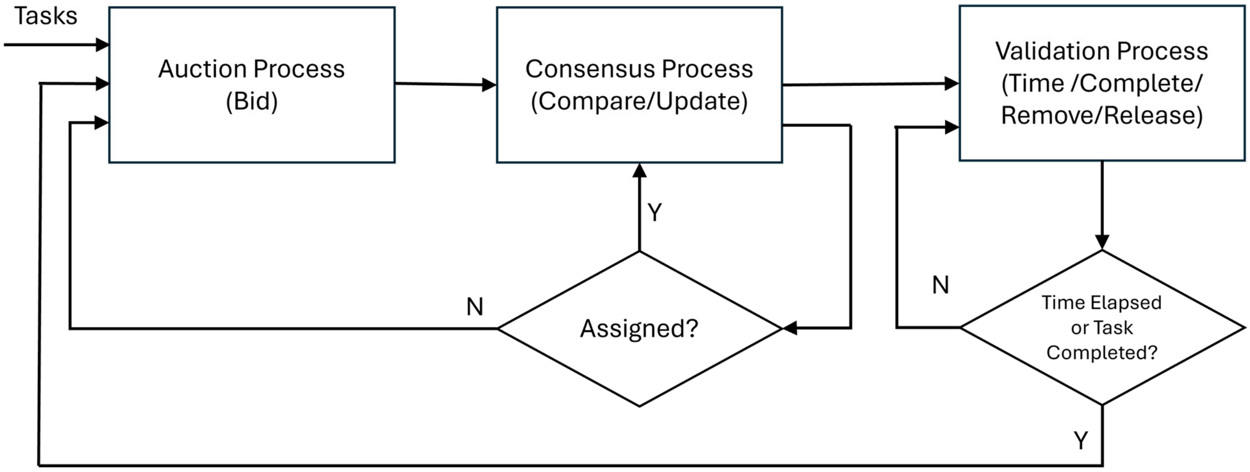


Figure 2. Caravan Auction (CarA) Algorithm Task Flow.

3.2. Auction Process

Each agent receives a graph $G = (V,E)$, which has a vector of the location of tasks $\boldsymbol{v}$ such that $\boldsymbol{v} \subset V$ and are the only global information provided to the agents. Each agent obtains a vector of scores $\mathbf{s}_i$ that serves as bids for possible tasks. Initialized as a set of zeros, this vector is populated based on the path-planning strategy, which provides values such that $s_{ij} \geq 0$.

The agents store five vectors of length N_m throughout the assignment process. The vector $\boldsymbol{x}_i(t) \in I$ is a list of agent assignments to tasks belonging to agent i at time t such that $x_{ij}(t) = k$ if task j is assigned to agent k and 0 otherwise. The index set was defined as $I \equiv \{1, \dots, N_u\}$. The following vector $\boldsymbol{t}_i(t) \in \mathbb{R}_+$ lists the times at which task j is assigned; this time vector, $\boldsymbol{t}_i(t)$, is initialized such that $t_{ij}(0) = 0\ \forall j$. The third vector is $\boldsymbol{y}_i(t) \in \mathbb{R}_+$, which provides the latest winning bid for each task. This is initialized as $y_{ij}(0) = 0\ \forall j$. The fourth vector $\boldsymbol{z}_i(t) \in \{0,1\}^{N_m}$ is the list of completed tasks. This vector is initialized as $z_{ij}(0) = 0\ \forall j$. The fifth and final vector of agent i, $\boldsymbol{f}_i(t) \in \{0,1\}^{N_m}$, is the list of released or orphaned tasks at time t. The value of f_{ij} is 1 if an agent assigned to task j exceeds its time limit to establish connectivity and 0 otherwise. The released task vector is initialized such that $f_{ij}(0) = 0\ \forall j$.

After the initialization of these vectors, the agent determines whether each task is unassigned or has been abandoned. If so, the agent determines whether it can accomplish the task. The ability of an agent to perform a certain task is expressed in the capability matrix $\boldsymbol{K}_i \in \{0,1\}^{N_u \times N_m}$ which defines whether an agent can perform a task. The capability element k_{ij} equals 1 if agent i can perform the task and 0 otherwise.

The winning bid list, list of completed tasks, and capability matrix are utilized to generate the available list of tasks $\boldsymbol{H}_i(t) \in \{0,1\}^{N_m}$ for agent i at time t by comparing the score it can achieve with the current winning bids on tasks that have not been completed, such that:

$$\boldsymbol{H}_i = \left(\boldsymbol{s}_i > \boldsymbol{y}_i(t) \wedge \neg \boldsymbol{z}_i(t)\right) \wedge \boldsymbol{K}_i \tag{14}$$

where $(a > b)$ is a Boolean vector whose j^{th} element is 1 if $a(j) > b(j)$ and 0 otherwise.

An unassigned agent selects task J_i and assigns it a maximum score based on the current winning bid list:

$$J_i = \underset{j}{argmax}\left\{H_{ij} s_{ij}\right\} \tag{15}$$

The agent then updates its task list $x_i(t)$ to show that it is now assigned task J_i. Time t was also recorded when task J_i was assigned. The agent then records the maximum score on the winning bid list for Task J_i. The competing agents also record the winning agent in their respective task lists and winning bid lists stored in their memory.

3.3. Consensus Process

Conflict resolution is necessary to ensure that the agent with the highest bid is assigned to the correct task. An underlying feature of this mitigation technique is the requirement for auction algorithms to be fully networked or to possess a way to route through agents to connect competitors. The CarA algorithm relaxes this condition by utilizing a consensus process that updates the bids of each agent and the winning list to determine the current winner for each task. Thus, synchronous bidding allows conflict resolution for all tasks, thereby eliminating the need for a rigid communication structure.

In this process, agents send their winning agent lists, winning bid lists, and assignment times to every agent with which they maintain an active connection. The communication matrix $G(t)$ behaves similarly to the adjacency matrix $G(t)$, in this algorithm. Agents with communication links between them are neighbors. Formally, if agents i and k are neighbors, $g_{ik}(t) = 1$. Additionally, every agent has a self-connecting link; therefore $g_{ii}(t) = 1\ \forall i$.

The lists are compared, and each task j is updated in the winning bid list for any incomplete task such that:

$$y_{ij}(t) = \max_k g_{ik}(t) \cdot y_{kj}(t-1) \tag{16}$$

A side effect of this dynamic behavior is that agent i may be reassigned using:

$$\begin{gathered} \psi_{ij}(t) = \underset{k}{argmax} g_{ik}(t) \cdot y_{k,J_i}(t-1) \\ x_{i,J_i} = \begin{cases} 0 & if\ \psi_{i,J_i} \neq i \\ x_{i,J_i}(t) & otherwise \end{cases} \\ t_{i,J_i} = \begin{cases} t_{k,J_i} & if\ \psi_{ij} \neq i \\ t_{i,J_i}(t) & otherwise \end{cases} \end{gathered} \tag{17}$$

Through information sharing, the winning bid list reflects the largest values of y_{ij} for agent i and its neighbors. Any agent relinquishing a task reenters the auction phase after visiting the validation phase. This could lead to repetitive cycling. A perturbation mechanism, ε, was added as a safety measure against churning behavior. The mechanism sets a buffer that a challenging agent must overcome to assume a task assignment. In this study, the perturbation mechanism was set as $\varepsilon = 10^{-6}$. Brunet [27] determined this value to be an appropriate measure to ensure fleet consensus. Therefore, the condition for agent i to swap tasks from agent k is as follows:

$$y_{ij} - y_{kj} > \varepsilon \tag{18}$$

In the event of a tie, resolution occurs based on the agent's lexicographical order.

3.4. Validation Process

The third phase of the CarA Algorithm is the validation phase, which involves agent monitoring and remediation. At onset, agent k will initialize its release times, $\boldsymbol{T_k} \in \mathbb{R}_+$ for each agent. Agent k sets the release time for agent i such that $T_{ki} = 0$.

Agents provide information regarding whether a task has been completed. Task completion is determined by the arrival of an agent at an assigned task location. Hence, if Agent i has the same location l as the assigned task l_{Ji}:

$$z_{ij}(t) = \begin{cases} 1 & \text{if } l_i(t) = l_{J_i}(t) \\ 0 & \text{if } l_i(t) \neq l_{J_i}(t) \end{cases} \tag{19}$$

The completion of a task resets all the specific task information and updates the completed task list. Deletion of a task involves agent i updating its stored vectors by resetting all elements associated with the completed task j to zero. These stored vectors represent the task time, score, winning agent, and winning bid.

As part of the validation phase, agent i notifies its neighbors of completing a task, such that:

$$z_{ij}(t) = g_{ik}(t) \cdot z_{kj}(t), \forall j \tag{20}$$

This communication provides decentralized shared memory to the agents of the task state even if there is no direct communication link between agent i and some other agent q; for example, if agent i can communicate with agent k but not with agent *q*. Information from agent *i* can still be provided to agent *q* assuming that agent *k* can relay the original message to agent *q*.

After updating its task list through its own efforts and information from other agents, agent *i* checks each task in the list for completed items marked for deletion, as described by:

$$z_{ij}(t) = 1, \forall j \tag{21}$$

The second major function of the validation process was to determine whether any task exceeded the allotted completion time. During each time step of the validation phase, Agent *k* attempts to communicate with agent *i* to update the release time T_{ki} such that:

$$T_{ki}(t) = g_{ik}(t) \cdot t_k + \delta_i, \forall i \tag{22}$$

where δ_i is the communication dropout time increment for agent *i*. This value represents the time at which an agent must report its status to its fellow agents. If an agent maintains active communication, release time will never be reached. If an agent loses connectivity (from its destruction or a major system failure), a release procedure is first examined to determine whether agent *k* is assigned task *j* or if task *j* is complete, such that:

$$x_{kj}(t) \neq 0 \land z_{kj}(t) = 0 \tag{23}$$

If agent *k* determines that the time limit to accomplish the assigned task J_i is exceeded, the agent will identify the task owner as agent *i* as described by:

$$x_{kj,J_i} | t_k > T_{ki} = i \tag{24}$$

Agent *k* then updates its records for task *j* by resetting the assignment time, score, task owner, and winning bid to zero, except for the lost task element, which is now identified as true. The determining agent *k* then notifies each agent that the task has been released, and the message is propagated as follows:

$$f_{ij}(t) = g_{ik}(t) \cdot f_{kj}(t), \forall j \tag{25}$$

Upon notification of a task's release, the agents compete for the orphaned task in the auction phase.

3.5. Path Planning Strategy

The proper routing of agents to tasks is necessary to provide a suitable approach for mobile agents. Although any number of path-planning strategies can be employed in ICF, this study examines a slight modification of Dijkstra's algorithm. In the original solution, Dijkstra's algorithm finds the shortest path between two points, producing the shortest path tree in time $O(n^2)$.

Under a constant rate of change, the agent will not only find the shortest path but also the shortest time between the two points. A minimization of the visit time t_v to the desired vertex results in maximization of the task score. The score for the k^{th} vertex is:

$$s_k = a_k e^{-\lambda t_v} \tag{26}$$

Reward a_k is for the task located at the k^{th} vertex. The discount factor for the score is defined as $0 \leq \lambda \leq 1$. (The undiscounted score is analogous to the reward described earlier.)

The agent determined the scores for each task located at the vertices in the graph. The score vector for agent *i*, $\mathbf{s}_i$, is a tuple of task scores along its optimal path. The vector is:

$$\boldsymbol{s_i} = \{\boldsymbol{s_k}, \boldsymbol{s_{k-1}}, \dots \boldsymbol{s_1}, \boldsymbol{s_0}\} \tag{27}$$

These scores act as bids during the auction phase of the CarA Algorithm.

4. Model Validation

The development of ICF provides a novel advancement that allows a fleet of UAVs to perform decentralized task assignment, given the random loss of UAVs in real-time. Next, the efficacy of this new approach was tested to determine the extent to which the UAV fleet had sufficient autonomy to avoid the need for human teleoperators (remote pilots) to make sudden decisions if many tasks become unassigned during a mission. The methodology used to test this new framework was divided into several areas: the model framework, research factors, and assessment parameters. The research factors provide variations in the environment, and solution approaches to test the effectiveness of the ICF under different constraints and against other benchmark approaches.

4.1. Model Framework

The UAVs were modeled by operating them in a two-dimensional environment. Abandoned tasks are presumed to occur when the UAV is lost. Lower-level UAV considerations were ignored, such as flight control and aerodynamic stability. Similarly, a zero-turning radius was provided for each vehicle. All the tasks were stationary. The real-world implications regarding fuel capacity were relaxed in the simulations. Obstacle avoidance, including UAV collisions, was ignored under the premise of staggered altitude levels for each UAV.

Several characteristics were used to define the model used in the study: sample size calculation, loss-of-agent strategy, scoring function, and randomization of agents and tasks.

A pilot study using a general full-factorial design was conducted to determine the sample size. The level of significance was set at 0.05, and because a high sensitivity was desired, β was set to 0.9. An analysis of the power curve from the pilot study indicated that the sample size should be 31 simulation runs per factor.

The Office of the Secretary of Defense released a study [57] on the reliability of RQ-2 Pioneer, RQ-5 Hunter, and RQ-1 Predator UAVs. The study reported mishap rates of over 100,000 flight hours, resulting in a severe loss of functionality, causing mission failure or complete destruction. The results showed that these UAVs had availabilities of 0.78, 0.98, and 0.93, respectively. Using a mean value of approximately 0.90, the deduced rate of vehicle loss was 0.10, and this study modeled a 10% reduction in the UAV fleet size. The loss time was randomly selected for each UAV. The set conditions for the pseudo-random number generator allow each simulation run for each algorithm to experience the same, although randomly generated, conditions.

The score for each agent after completing the task was set using a predefined function. This function was based on the task reward, discount factor, and time for task completion. The task reward, *a,* was fixed at an arbitrary score of 1000. The reward was discounted based on the cost of traveling to each point. The discount factor, λ, was set at 0.1, for all task score calculations. The score *s* is the exponential decay of the reward based on the time *t*:

$$s = a\lambda^{-t} \tag{28}$$

Therefore, the longer it takes to execute a task, the lower the score will be. The distances between the agents and tasks are critical for the investigation of performance. To mitigate bias for any specific solution approach, the locations of both the agents and tasks were randomized. A uniform distribution was used to ensure that the probabilities of occurrence along the intervals were equal. Each combination of factor levels was subjected to the same randomization sequence to ensure repeatable conditions.

4.2. Research Factors

The proposed research factors investigated how well the ICF scales, its robustness under communication constraints, and a general comparison with the existing approaches. The research factors in this study were the number of agents and tasks, communication range, and solution approach.

Scaling is defined as the number of agents and tasks required for each simulation. Although an infinite combination of tasks and agents is possible, Brunet [35] empirically determined that the worst case for a solution approach such as ICF is when the number of agents and tasks is the same. Therefore, this convention was used in this study. Three levels were provided: small-scale (five agents and five tasks), medium-scale (15 agents and 15 tasks), and large-scale (30 agents and 30 tasks).

The communication range represents the maximum distance between two agents, allowing information exchange. Three transmit/receive levels were examined for this factor: low (0.1*D*), medium (0.5*D*), and high (1*D*), where *D* is the diagonal map length.

In addition to comparing its own performance over various levels of scaling and communication range, ICF is benchmarked against two established solution approaches: the Implicit Coordination Algorithm (ICA) [58] and the Greedy-based Auction Algorithm (GBAA) [35]. The solution approaches differ in style and offer unique comparisons. ICA is a decentralized task assignment approach that examines all possibilities to provide an optimal solution. GBAA is a decentralized task-assignment solution approach that uses auction algorithms and adjudicates conflicts with direct neighbors only. Therefore, members who do not have a direct link with another agent for any reason, such as distance, cannot resolve task conflicts.

4.3. Assessment Parameters

The optimality gap, convergence steps, and execution time were evaluated to assess the performance of ICF, ICA, and GBAA. The optimality gap is a relative comparison of closeness to the best possible answer. The convergence steps indicate how long it takes for the UAV fleet to reach a consensus regarding task assignment. Specifically, the number of steps was defined as the number of iterations required until each UAV could not outbid another UAV for task ownership. The execution time is the output variable that determines how quickly the UAV fleet accomplishes its task. Execution time $\bar{t}$, is defined as the amount of time it takes for the last task to be accomplished in any given simulation:

$$\bar{t} = max\{t: \boldsymbol{z}(t) = \mathbf{1}\} \tag{29}$$

where $\mathbf{z}(t)$ is the vector of completed tasks, such that 1 is a completed task, and 0 otherwise. Execution times are important when considering real-world effects, as they describe when the fleet has completed all assigned tasks.

5. Results of Numerical Analysis

A numerical study was conducted using the model framework provided in Section 4.1. Specifically, this study attempts to characterize the ICF by answering several questions. Does this approach make a novel contribution through the reallocation of orphaned tasks using auction and consensus algorithms for a decentralized fleet? Does the approach match the performance of the benchmark algorithms? Can it be scaled up appropriately using more agents and tasks? How does communication affect performance?

To ensure a well-balanced experiment, the Monte Carlo simulations used tenets of randomization and replication. The use of randomization, particularly for the locations of agents, locations of tasks, and time of vehicle loss, prevents bias towards any one algorithm. Replication was used to reduce the variability in the experimental results.

Analyses of the parameters of interest were collected from the simulations to draw conclusions regarding the behavior of the ICF. The mean and standard deviation of the output variables were reported for each solution approach. Additionally, an inferential strategy was deployed through a full factorial experimental design and general linear model fit. Analysis of variance (ANOVA) and Tukey's range tests were used to assess the interactions between variables and differences in means.

5.1. Results of Descriptive Statistics

A full factorial design of 837 Monte Carlo simulations was performed. The novelty of ICF was determined by inspecting the behavior of the UAV fleet throughout the simulations. Each run was assessed using the following three questions:

1. Are the tasks assigned during the simulation run?
2. Did the algorithm remove the tasks after an agent failed to report its status after a predefined time limit?
3. Did the agents validate a task as completed after its execution?

The results confirmed that these three questions were answered affirmatively, providing strong evidence that the ICF offers a novel approach to task reassignment of a degraded UAV fleet. Descriptive statistics were conducted on the single-assignment problem to determine the ICF behavior. A series of tables is presented to show the behavior exhibited by each solution approach over the communication ranges with respect to the parameters of interest.

Table 1 shows that ICF outperformed both ICA and GBAA at a low transmit/receive factor level in terms of the optimality gap mean for a small-scale environment. The medium and high transmit/receive levels show that ICF is outperformed by ICA, on average. However, ICF outperformed GBAA in both cases when the optimality gap means were compared. The variability of the ICF in terms of optimality showed mixed results compared to the benchmarks. An increase in communication length further stabilized the variance in the ICF. For a medium-scale environment, ICF outperformed ICA and was comparable to GBAA in terms of the low transmit/receive factor level when viewing the optimality gap. The general trend is that ICF underperforms in terms of the optimality gap and stabilization against benchmark algorithms when considering the optimality gap. For a large-scale environment, ICF is outperformed by ICA and GBAA when comparing the optimality gap means over the transmit/receive factor levels, and ICF exhibits more variance in performance.

Table 1. Optimality Gap *vs.* Communication Length by Solution Method & Task Size.

Solution	**Number of Tasks & Agents**	**0.1D**		**0.5D**		**1D**	
		$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
ICA	**5 & 5**	0.316	0.202	0.158	0.120	0.145	0.120
	15 & 15	0.274	0.078	0.096	0.095	0.080	0.045
	30 & 30	0.232	0.051	0.079	0.036	0.071	0.029
GBAA	**5 & 5**	0.292	0.147	0.179	0.103	0.171	0.096
	15 & 15	0.246	0.087	0.125	0.127	0.109	0.055
	30 & 30	0.256	0.042	0.104	0.027	0.090	0.027
ICF	**5 &5**	0.237	0.171	0.174	0.148	0.164	0.146
	15 & 15	0.247	0.114	0.133	0.135	0.117	0.081
	30 & 30	0.255	0.057	0.122	0.048	0.110	0.043

Table 2 provides a comparison of the effects of the solution approach on the fleet's convergence steps across three different environmental scales. Across all three solution methods, the number of convergence steps increased as the combination of task and fleet size increased. Notably, the increase in ICF and GBBA was much less than that in ICA. In a small-scale environment, ICF consistently converges faster than ICA but not GBAA over the communication ranges tested. Similarly, ICF has more stable results for the convergence steps than ICA, but has less consistency than GBAA. The ICF shows similar behavior within the medium-scale environment compared with the other two solution approaches, with ICF outperforming ICA in terms of convergence steps and deviation. GBAA exhibited the best performance in terms of both the statistical measures. In a large-scale environment, ICF outperformed ICA in terms of convergence step mean and variance for all communication ranges. GBAA generally outperforms ICF in terms of both convergence step mean and deviation. Yet ICF provides more stability in the convergence steps for a high transmit/receive factor level.

Table 2. Convergence Steps *vs.* Communication Length by Solution Method & Task Size.

Solution	**Number of Tasks & Agents**	**0.1D**		**0.5D**		**1D**	
		$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
ICA	**5 & 5**	24.16	9.72	41.23	11.19	43.23	12.22
	15 & 15	208.03	60.13	812.16	101.04	816.29	110.62
	30 & 30	1069.35	263.80	4938.42	449.05	4964.52	504.31
GBAA	**5 & 5**	7.42	0.56	9.55	1.03	9.90	1.04
	15 & 15	18.55	0.57	26.29	2.21	30.10	2.87
	30 & 30	35.42	1.18	50.68	4.55	57.74	5.40
ICF	**5 &5**	16.81	3.65	24.81	6.80	13.26	2.92
	15 & 15	60.87	17.23	128.19	71.79	37.23	3.29
	30 & 30	134.48	50.84	393.45	204.17	70.10	1.92

Table 3 lists the impact of various solution approaches on the execution times as the communication lengths and task/agent combinations are varied. In a small-sized environment, ICF provides faster execution times across each communication level. The standard deviations for ICF are also smaller and thus represent a more stable set of results than the benchmark algorithms, with one exception. At a high transmit/receiver factor level, ICA had a lower standard deviation than ICF.

In a medium-sized environment, GBAA has the lowest average execution time for the low transmit/receive level, with ICF and ICA as the second and third, respectively. For the medium and high transmit/receive levels, ICF had the lowest mean execution times, whereas GBAA had the lowest standard deviations for the low and medium transmit/receive levels. ICF exhibited the lowest standard deviation for execution times at a high transmit/receive level. In a large-scale environment, the results are similar to those of GBBA, achieving faster execution times than ICF and ICA, whereas ICF achieves the lowest standard deviation for execution times under the high transmit/receive level.

Table 3. Execution Times *vs.* Communication Length by Solution Method & Task Size.

Solution	Number of Tasks & Agents	0.1D		0.5D		1D	
		$\bar{x}$	*s*	$\bar{x}$	*s*	$\bar{x}$	*s*
ICA	**5 & 5**	9.04	2.56	12.55	3.29	12.17	3.27
	15 & 15	8.96	2.17	15.90	4.62	15.12	4.76
	30 & 30	7.85	2.22	14.80	4.80	15.23	5.06
GBAA	**5 & 5**	6.95	2.50	12.23	3.91	13.01	3.86
	15 & 15	5.07	1.37	13.03	3.17	17.38	5.26
	30 & 30	4.40	1.26	12.37	2.26	17.71	4.32
ICF	**5 &5**	6.29	2.19	9.99	3.06	10.94	3.45
	15 & 15	5.50	1.60	11.98	3.57	13.54	3.58
	30 & 30	5.28	1.41	12.71	3.23	14.08	3.57

5.2. Results of Inferential Statistics

Table 4 presents the ANOVA results of the full factorial analysis with respect to the optimality. The most interesting aspect of this table is the difference between the solution approaches when the response variable is the optimal gap. The table shows that the solution approaches appear to have little significant difference in their behavior in terms of the optimality gap. The source of variation is broken down by the solution approach (S), the scale of the number of agents and tasks (AT), the communication length (CL), and the set of 2 and 3 combination interactions. The combinations of factors for testing interactions are denoted with an asterisk (*) between factors. The solution approach includes the ICF, ICA, and GBBA. The scale of agents vs. tasks includes small (5 × 5), medium (15 × 15), and large (30 × 30). The communication length includes 0.1D, 0.5D, and 1D compared to the communication distance (D).

Table 4. ANOVA of All Solutions for Optimality Gap.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
S	2	49.72	4.05	2.02	1.89	0.152
AT	2	69.00	44.32	22.16	20.69	0.000
CL	2	1080.05	337.18	168.59	157.42	0.000
S*AT	4	4.49	7.02	1.76	1.64	0.162
S*CL	4	7.69	10.76	2.69	2.51	0.041
AT*CL	4	2.16	3.23	0.81	0.75	0.556
S*AT*CL	8	7.65	7.65	0.96	0.89	0.522
Error	810	867.49	867.49	1.07		
Total	836	2088.24				

Table 5 presents the ANOVA results of the full factorial analysis with respect to the optimality. The most interesting aspect of this table is the difference between the solution approaches when the response variable is the optimal gap. The table shows that the solution approaches appear to have little significant difference in their behavior in terms of the optimality gap.

Table 5. ANOVA of All Solutions for Convergence Steps.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
S	2	5.878	5.704	2.8522	11,289.98	0.000
AT	2	5.645	5.723	2.8617	11,327.81	0.000
CL	2	0.307	0.313	0.1567	620.29	0.000
S*AT	4	0.467	0.463	0.1159	458.56	0.000
S*CL	4	0.353	0.354	0.0884	349.89	0.000
AT*CL	4	0.012	0.012	0.0030	11.96	0.000
S*AT*CL	8	0.019	0.019	0.0024	9.59	0.000
Error	800	0.202	0.202	0.0003		
Total	826	12.883				

Table 6 presents the ANOVA results for the execution times of the solution approach. The table indicates that the solution approaches vary significantly in terms of execution times. The results of the Tukey's range test show that each solution approach differs significantly from the others, with ICF offering the fastest execution time.

Table 6. ANOVA of All Solutions for Execution Time.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
S	2	19.22	19.22	9.61	40.03	0.000
AT	2	5.28	5.28	2.64	11.00	0.000
CL	2	237.75	237.75	118.87	495.11	0.000
S*AT	4	1.27	1.27	0.32	1.33	0.258
S*CL	4	16.72	16.72	4.18	17.40	0.000
AT*CL	4	16.90	19.90	4.23	17.60	0.000
S*AT*CL	8	2.15	2.15	0.27	1.12	0.346
Error	810	194.48	194.48	0.24		
Total	836	493.77				

6. Conclusions

In the numerical study, using statistics, CarA and ICF algorithms were tested for their novelty and efficacy. Under 837 Monte Carlo simulations, the ICF must answer several questions that previous solution approaches did not. The descriptive statistics show that the ICF exhibits competitive performance relative to benchmark algorithms. Additionally, the ICF remained close to that of the benchmark algorithms in terms of the optimality gap. Evaluating the number of convergence steps, ICF offers improvements over ICA but does not reach the GBAA level. In contrast, the ICF produces means and standard deviations that are generally better than those of ICA and GBAA.

The results of the inferential statistical analysis provide performance insights, with the ICF displaying no significant difference between the other solution approaches in terms of the optimality gap. The ANOVA for ICF concludes that there is no optimality gap in behavioral performance with increases in agents and tasks. Significance in terms of the optimality gap is shown between low/medium communication ranges but not between medium/high ranges. The results indicate that ICF converges in fewer steps than ICA, but does not confirm faster convergence than GBAA, and ICF shows slower convergence with an increase in scaling. Nonlinear behavior appears in the convergence steps for the ICF with an increase in communication length. Additionally, ICF provided significantly lower execution times than the benchmark algorithms. An increase from low to medium scaling of agents and tasks resulted in higher execution times. However, this behavior was not observed for medium-to large-scale increases.

Considering the significant differences in the convergence steps, modifications to the ICF will include a "not to exceed" number for convergence opportunities that align with the GBAA approach. This would limit computational costs. This would also allow an opportunity to determine in a more "apples to apples" scenario whether this creates a significant difference in terms of optimality and execution times. Another modification would be to only allow for negotiations among platforms that have never acquired a task. This would reduce the number of convergence steps. This could potentially have an impact on global optimization, so it will need to be explored in future work.

A second area of future work concerns comparisons with the traditional Consensus-based Bundle Auction Algorithm (CBBA). The CarA algorithm expands on the CBBA concept by allowing a replan feature in the event of a lost vehicle, which the traditional CBAA does not do. The CarA algorithm should be dominant in terms of optimality as some of the tasked assets using CBAA might not complete their missions. This is expected to result in lower scores for CBBA than CarA, which has a chance to be re-evaluated and a higher scoring target selected.

This study introduces a novel framework that allows a fleet of decentralized agents to determine when an agent has been lost and how to best reassign its orphaned tasks. This study provides insight into the broader conceptual question of whether autonomous advancements can alleviate teleoperators' cognitive load when UAVs are in a high-risk, real-time environment. An analysis of the proposed CarA Algorithm and ICF indicates that this new solution approach has the necessary characteristics to alleviate cognitive loading and stress on a teleoperator by providing a reconfiguration of tasks resulting from lost UAVs. Statistical analyses proved that ICF can provide performance comparable to existing solutions in scoring, scalability, and execution. Analyses of the robustness of the ICF demonstrated stable expectations with respect to the scaling effects and communication limitations. Through the utilization of auction and consensus algorithms, this study contributes to research and practice by providing teleoperators and decision makers with a new approach offering autonomous vehicle behavior, applicable in situations where overall network resiliency and performance are mandated.

Author Contributions

Conceptualization, W.T.B.; Methodology, W.T.B., G.M.N.; Software, W.T.B.; Validation, W.T.B., G.M.N., C.D.M. and A.Y.O.; Formal Analysis, W.T.B.; Investigation, W.T.B., G.M.N.; Resources, W.T.B.; Data Curation, W.T.B.; Writing—Original Draft Preparation, W.T.B., G.M.N.; Writing—Review & Editing, W.T.B., G.M.N., C.D.M. and A.Y.O.; Visualization, W.T.B.; Supervision, G.M.N.; Project Administration, W.T.B.; Funding Acquisition, W.T.B.

Ethics Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Research data is available by request to the corresponding author.

Funding

This research was funded in part by the United States Department of Defense, Naval Information Warfare Center Atlantic.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Khan MA, Ectors W, Bellemans T, Janssens D, Wets G. UAV-based traffic analysis: A universal guiding framework based on literature survey. *Transp. Res. Proc.* **2017**, *22*, 541–550. doi:10.1016/j.trpro.2017.03.043.
2. Szóstak M, Nowobilski T, Mahamadu A-M, Pérez DC. Unmanned aerial vehicles in the construction industry—Towards a protocol for safe preparation and flight of drones. *Int. J. Intell. Unman Syst.* **2023**, *11*, 296–316. doi:10.1108/IJIUS-05-2022-0063.
3. Lei X, Hu X, Wang G, Luo H. A multi-UAV deployment method for border patrolling based on Stackelberg game. *J. Syst. Eng. Electron.* **2023**, *34*, 99–116. doi:10.23919/JSEE.2023.000022.
4. Mori K, Nishira M, Nishikawa H, Tomiyama H. Exact and heuristic approaches to surveillance routing with a minimum number of drones. *Drones Auton. Veh.* **2024**, *1*, 1004. doi:10.35534/dav.2024.10004.
5. Hu J, Yang J. Application of distributed auction to multi-UAV task assignment in agriculture. *Int. J. Precis. Agric. Aviat.* **2018**, *1*, 44–50. doi:10.33440/j.ijpaa.20180101.0008.
6. Guebsi R, El Wai R. Leveraging drone technology for precision agriculture: A comprehensive case study in Sidi Bouzid, Tunisia. *Drones Auton. Veh.* **2025**, *2*, 10006. doi:10.70322/dav.2025.10006.
7. Saunders J, Saeedi S, Li W. Autonomous aerial robotics for package delivery: A technical review. *J. Field Robot.* **2024**, *41*, 3–49. doi:10.1002/rob.22231.
8. Sorbelli FB. UAV-based delivery systems: A systematic review, current trends, and research challenges. *J. Auton. Transp. Syst.* **2024**, *1*, 1–40. doi:10.1145/3649224.
9. Nishira M, Nishikawa H, Kong X, Tomiyama H. An integer programming approach to multi-trip routing of delivery drones at load-dependent flight speed. *Drones Auton. Veh.* **2024**, *1*, 10008. doi:10.35534/dav.2024.10008.
10. Patrinopoulou N, Daramouskas I, Meimetis D, Lappas V, Kostopoulos V. A distributed framework for persistent wildfire monitoring with fixed wing UAVs. *Drones Auton. Veh.* **2024**, *1,* 10009. doi:10.35534/dav.2024.10009.
11. Jan GE, Lei T, Sun CC, You ZY, Luo C. On the problems of drone formation and light shows. *IEEE Trans. Consum. Electron.* **2024**, *70*, 5259–5268. doi:10.1109/TCE.2024.3421516.
12. Wang G, Wang F, Wang J, Li M, Gai L, Xu D. Collaborative target assignment problem for large-scale UAV swarm based on two-stage greedy auction algorithm. *Aerosp. Sci. Technol.* **2024**, *149*, 109–146. doi:10.1016/j.ast.2024.109146.

13. Dantzig GB, Ramser JH. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. doi:10.1287/mnsc.6.1.80.
14. Psaraftis HN. Dynamic vehicle routing problems. In *Vehicle Routing: Methods and Studies*; Golden BL, Assad AA, Eds.; Elsevier: Amsterdam, The Netherlands, 1988; pp. 223–247.
15. Chiang T-C, Hsu W-H. A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows. *Comput. Oper. Res.* **2014**, *45*, 25–37. doi:10.1016/j.cor.2013.11.014.
16. Archetti C, Corberán Á, Plana I, Sanchis JM, Speranza MG. A branch-and-cut algorithm for the orienteering arc routing problem. *Comput. Oper. Res.* **2016**, *66*, 95–104. doi:10.1016/j.cor.2015.08.003.
17. Montemanni R, Weyland D, Gambardella LM. An enhanced ant colony system for the team orienteering problem with time windows. In Proceedings of the International Symposium on Computer Science and Society, Kota Kinabalu, Malaysia, 16–17 July 2011; pp. 381–384. doi:10.1109/ISCCS.2011.95.
18. Evers L, Glorie K, van der Ster S, Barros AI, Monsuur H. A two-stage approach to the orienteering problem with stochastic weights. *Comput. Oper. Res.* **2014**, *43*, 248–260. doi:10.1016/j.cor.2013.09.011.
19. Moon S, Oh E, Shim DH. An integral framework of task assignment and path planning for multiple unmanned aerial vehicles in dynamic environments. *J. Intell. Robot. Syst.* **2013**, *70*, 303–313. doi:10.1007/s10846-012-9740-3.
20. Góral I, Tchoń K. Lagrangian jacobian motion planning: A parametric approach. *J. Intell. Robot. Syst.* **2017**, *85*, 511–522. doi:10.1007/s10846-016-0394-4.
21. Zengin U, Dogan A. Cooperative target pursuit by multiple UAVs in an adversarial environment. *Robot. Auton. Syst.* **2011**, *59*, 1049–1059. doi:10.1016/j.robot.2011.08.006.
22. Frontera G, Martín DJ, Besada JA, Gu D-W. Approximate 3D euclidean shortest paths for unmanned aircraft in urban environments. *J. Intell. Robot. Syst.* **2017**, *85*, 353–368. doi:10.1007/s10846-016-0409-1.
23. Hota S, Ghose D. Waypoint-based trajectory planning of fixed-wing MAVs in 3D space. *J. Intell. Robot. Syst.* **2017**, *86*, 95–113. doi:10.1007/s10846-016-0415-3.
24. Koubâa A, Cheikhrouhou O, Bennaceur H, Sriti M-F, Javed Y, Ammar A. Move and improve: a market-based mechanism for the multiple depot multiple travelling salesmen problem. *J. Intell. Robot. Syst.* **2017**, *85*, 307–330. doi:10.1007/s10846-016-0400-x.
25. Hino T, Tsuchiya T. Heuristic path planning of unmanned aerial vehicle formations. *Int. J. Intell. Unman Syst.* **2013**, *1*, 121–144. doi:10.1108/20496421311330056.
26. Huo J, Zenkevich SL, Nazarova AV, Zhai M. Path planning based on map matching in UAV/UGV collaboration system. *Int. J. Intell. Unman Syst.* **2021**, *9*, 81–95. doi:10.1108/IJIUS-03-2019-0020.
27. Cheng H, Page J, Olsen J. Cooperative control of UAV swarm via information measures. *Int. J. Intell. Unman Syst.* **2013**, *1*, 256–275. doi:10.1108/IJIUS-01-2013-0001.
28. Taran B, Homaeinezhad M. Fully- and partially- distributed adaptive consensus of second-order multi-agent systems using only relative position measurements. *Drones Auton. Veh.* **2024**, *1*, 10014. doi:10.70322/dav.2024.10014.
29. Chen J, Du C, Zhang Y, Han P, Wei W. A clustering-based coverage path planning method for autonomous heterogeneous UAVs. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25546–25556. doi:10.1109/TITS.2021.3066240.
30. Shetty VK, Sudit M, Nagi R. Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. *Comput. Oper. Res.* **2008**, *35*, 1813–1828. doi:10.1016/j.cor.2006.09.013.
31. Nanjanath M, Gini M. Repeated auctions for robust task execution by a robot team. *Robot. Auton. Syst.* **2010**, *58*, 900–909. doi:10.1016/j.robot.2010.03.011.
32. Buer T, Kopfer H. A Pareto-metaheuristic for a bi-objective winner determination problem in a combinatorial reverse auction. *Comput. Oper. Res.* **2014**, *41*, 208–220. doi:10.1016/j.cor.2013.04.004.
33. Duan X, Liu H, Tang H, Cai Q, Zhang F, Han X. A novel hybrid auction algorithm for multi-UAVs dynamic task assignment. *IEEE Access* **2020**, *8*, 86207–86222. doi:10.1109/ACCESS.2019.2959327.
34. Izhboldina V, Lebedev I. Group movement of UAVs in environment with dynamic obstacles: A survey. *Int. J. Intell. Unman Syst.* **2023**, *11*, 268–284. doi:10.1108/IJIUS-06-2021-0038.
35. Brunet L. Consensus-Based Auctions for Decentralized Task Assignment. Masters Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2008. Available online: http://hdl.handle.net/1721.1/44926 (accessed on 01 September 2025).
36. Alighanbari M. Robust and Decentralized Task Assignment Algorithms for UAVs. Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2007. Available online: http://dspace.mit.edu/handle/1721.1/7582 (accessed on 01 September 2025).
37. Choi H-L, Brunet L, How JP. Consensus-based decentralized auctions for robust task allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–926. doi:10.1109/TRO.2009.2022423.
38. Zhang Y, Feng W, Shi G, Jiang F, Chowdhury M, Ling SH. UAV swarm mission planning in dynamic environment using Consensus-Based Bundle Algorithm. *Sensors* **2020**, *20*, 2307. doi:10.3390/s20082307.

39. Chen J, Qing X, Ye F, Xiao K, You K, Sun Q. Consensus-based bundle algorithm with local replanning for heterogeneous multi-UAV system in the time-sensitive and dynamic environment. *J. Supercomput.* **2022**, *78*, 1712–1740. doi:10.1007/s11227-021-03940-z.
40. Jung S. Real-time UAV Autonomy through Offline Calculations. Masters Thesis, Purdue University, West Lafayette, Indiana, 2010.
41. Li W, Ren R, Shi M, Lin B, Qin K. Seeking secure adaptive distributed discrete-time observer for networked agent systems under external cyber attacks. *IEEE Trans. Consum. Electron.* **2025**, *71*, 918–930. doi:10.1109/TCE.2025.3535083.
42. Bertsekas DP. Auction algorithms. In *Encyclopedia of Optimization*; Pardalos PM, Ed.; Springer: Boston, MA, USA, 2001; pp. 73–77.
43. Rabbath CA. A finite-state machine for collaborative airlift with a formation of unmanned air vehicles. *J. Intell. Robot. Syst.* **2013**, *70*, 233–253. doi:10.1007/s10846-012-9692-7.
44. Li W, Yue J, Shi M, Lin B, Qin K. Neural network-based dynamic target enclosing control for uncertain nonlinear multi-agent systems over signed networks. *Neural Netw.* **2025**, *184*, 107057. doi:10.1016/j.neunet.2024.107057.
45. Campion M, Ranganathan P, Faruque S. UAV swarm communication and control architectures: A review. *Unmanned Veh. Syst.* **2018**, *7*, 93–106. doi:10.1139/juvs-2018-0009.
46. Tahir A, Böling J, Haghbayan MH, Toivonen HT, Plosila J. Swarms of unmanned aerial vehicles—A survey. *J. Ind. Inf. Integr.* **2019**, *16*, 100106. doi:10.1016/j.jii.2019.100106.
47. Kada B, Khalid M, Shaikh MS. Distributed cooperative control of autonomous multi-agent UAV systems using smooth control. *J. Syst. Eng. Electron.* **2020**, *31*, 1297–1307. doi:10.23919/JSEE.2020.000100.
48. Adoni WYH, Lorenz S, Fareedh JS, Gloaguen R, Bussmann M. Investigation of autonomous multi-UAV systems for target detection in distributed environment: Current developments and open challenges. *Drones* **2023**, *7*, 263. doi:10.3390/drones7040263.
49. Javed S, Hassan A, Ahmad R, Ahmed W, Ahmed R, Saadat A, et al. State-of-the-art and future research challenges in UAV swarms. *IEEE Internet Things J.* **2024**, *11*, 19023–19045. doi:10.1109/JIOT.2024.3364230.
50. Alqudsi Y, Makaraci M. UAV swarms: research, challenges, and future directions. *J. Eng. Appl. Sci.* **2025**, *72*, 12. doi:10.1186/s44147-025-00582-3.
51. Kourtis MA, Xilouris G, Oikonomakis A, Batistatos MC, Chochliouros I. Integration of Drone Connectivity in 5G: An Examination of the OASEES Framework. In Proceedings of the 2023 IEEE Future Networks World Forum (FNWF), Baltimore, MD, USA, 13–15 November 2023; pp. 1–6; doi:10.1109/FNWF58287.2023.10520379.
52. Amponis G, Lagkas T, Zevgara M, Katsikas G, Xirofotos T, Moscholios I, et al. Drones in B5G/6G networks as flying base stations. *Drones* **2022**, *6*, 39. doi:10.3390/drones6020039.
53. Jagatheesaperumal SK, Rahouti M, Chehri A, Xiong K, Bieniek J. Blockchain-based security architecture for unmanned aerial vehicles in B5G/6G services and beyond: A comprehensive approach. *IEEE Open J. Commun. Soc.* **2025**. doi:10.1109/OJCOMS.2025.3528220.
54. Hartmann K, Steup C. The vulnerability of UAVs to cyber attacks—An approach to the risk assessment. In Proceedings of the 5th International Conference on Cyber Conflict, Tallinn, Estonia, 4–7 June 2013; pp. 1–23. Available online: https://ieeexplore.ieee.org/abstract/document/6568373 (accessed on 01 September 2025).
55. Barnawi WT. Integrated Consensus-Based Frameworks for Unmanned Vehicle Routing and Targeting Assignment. Doctoral Dissertation, The University of Alabama in Huntsville, Huntsville, AL, USA, 2015. Available online: https://www.proquest.com/openview/52c5bf8a45854a246f9471cfca26bb53/1?cbl=18750&pq-origsite=gscholar (accessed on 01 September 2025).
56. Beard R, Stepanyan V. Synchronization of information in distributed multiple vehicle coordinated control. In Proceedings of the IEEE Conference on Decision and Control, Maui, HI, USA, 9–12 December 2003; pp. 2029–2034. doi:10.1109/CDC.2003.1272913.
57. Office of the Secretary of Defense. Unmanned Aerial Vehicle Reliability Study. 2003, Unpublished manuscript.
58. Alighanbari M, How JP. Decentralized task assignment for unmanned aerial vehicles. In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 15 December 2005; pp. 5668–5673. doi:10.1109/CDC.2005.1583066.