ORIGINAL RESEARCH Open access

Decentralized Cooperative Behaviors for Multi-Agent Simultaneous Search and Routing

Thomas Anderson[1*], Maria Silva[2], James Walker[3], Emma Wilson[4]

Abstract

Utilising autonomous capabilities enables Unmanned Aerial Vehicles (UAVs) to tackle diverse multi-agent problems in practical settings, including search-and-rescue operations and surveillance activities. In such contexts, the overarching goal is typically better achieved when the challenge is optimally divided among the participating agents. Yet, within uncertain, constantly changing, and frequently only partially observable surroundings, approaches relying on centralised global optimisation cannot be implemented. Consequently, agents must rely on their individual understanding of the environment, making optimal independent choices that may, at times, prove limited in scope. Given that numerous agents operate without central direction, the question arises of how to minimise rivalry and encourage joint efforts. This study focuses on the scenario in which several UAVs simultaneously search for tasks in a shared space, determining efficient paths to those tasks and visiting them. The research is inspired by the notion that teamwork can remain uncomplicated and occur without direct conversation, achieved instead through the thoughtful construction of each agent's decision-making rules. By emphasising the aspects of information exchange, the applicability of one core agent strategy is extended. With only slight alterations, this strategy can yield agents with independent, cooperative, or competitive tendencies. More precisely, analysis of sensor and communication ranges reveals that expanding sensor coverage may actually reduce overall effectiveness, whereas modelling the probable goals of nearby agents offers a significantly superior method.

Keywords Unmanned aerial vehicles (UAVs), Path planning, Search and rescue, Multi-agent systems, Travelling salesman problem

*Correspondence:
Thomas Anderson
thomas.anderson@food.ku.dk
[1] Department of AI for Precision Root Crop Harvesting and Soil Compaction Mapping, Faculty of Agriculture, University of Copenhagen, Copenhagen, Denmark
[2] Department of Machine Learning for Drought Stress Detection in Soybean using Hyperspectral Imaging, Faculty of Agriculture, University of São Paulo, São Paulo, Brazil
[3] Department of Smart Autonomous Spraying Drones for Orchards, Faculty of Agriculture, University of Edinburgh, Edinburgh, United Kingdom
[4] Department of Precision Irrigation Scheduling based on Real-Time Evapotranspiration, Faculty of Agriculture, University of Melbourne, Melbourne, Australia

Introduction

Unmanned Aerial Vehicles (UAVs) offer a promising and innovative means of addressing many practical real-world challenges, such as reconnaissance and surveillance [1-7], search and rescue [8-10], and package delivery [11, 12]. Employing several UAVs concurrently can resolve these challenges more quickly, with greater efficiency and enhanced reliability [13] compared to standard manned alternatives. Developing models and plans for multi-agent situations often proves demanding owing to a quickly enlarging set of possible decisions, further complicated by interactions among agents and between agents and their surroundings. Moreover, this situation frequently demands coordination and

This is an open access article distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0), which permits non-commercial use, distribution, reproduction, and adaptation in any medium or format, provided that appropriate credit is given to the original author(s) and source, a link to the license is provided, and any changes made are indicated. Adapted material must be distributed under the same license.

© 2026 The Author(s). | Published by Agro Intelligence Research Press

information sharing that might prove unfeasible under many conditions [14, 15]. A large proportion of UAV systems and readily available commercial options are created separately and usually support solely single-agent operations. Without agents being purpose-built for multi-agent contexts or subject to central oversight, uniformity in behaviour may become unavoidable.

This paper explores the situation of a collection of identical UAVs that locate, navigate toward, and reach various positions (tasks) scattered across the environment. The core idea is that effective teamwork remains possible even in the absence of overt communication, provided the agents' standalone decision-making processes are carefully crafted. In particular, attention to the type of information passed between agents or to the assumptions agents form about others' likely moves can yield quite different results. These results can show independent, cooperative, or competitive patterns. Specifically, examination of sensor and communication range effects demonstrates that heightened awareness of the environment state, gained via broader communication or sight distance, can in reality impair collective outcomes. A more measured handling of available data, achieved by basic estimation of nearby agents' likely objectives, turns out to be considerably more productive.

This paper opens with a definition of the Multi-Agent Simultaneous Searching and Routing Problem (MSSRP), including the complete global optimisation formulation, followed by a detailed explanation of how the current challenge diverges from that ideal case and the suggested heuristic method for producing workable solutions. Subsequently, presents four distinct agent strategies derived from limited changes to one foundational behaviour. Also details the multi-agent simulation platform created to assess these strategies. The same platform is then applied to examine how different selections of agent parameters affect overall system results.

The Multi-Agent Simultaneous Searching and Routing Problem

The Multi-Agent Simultaneous Searching and Routing Problem (MSSRP) investigated here is the task of locating tasks within an environment while simultaneously determining routes among them. In practice, this means each agent persistently pursues its current optimal path linking the tasks it has identified, updating it whenever new tasks become known. One critical difficulty when several agents are involved is the substantial waste that can result when agents compete for the same tasks. Under a fully centralised arrangement with complete awareness of all task positions, every task could be assigned to the most efficient individual UAVs from the outset. In contrast, the decentralised setting examined here features several notable distinctions: (1) UAVs identify tasks only by traversing the space and reaching a distance close enough to detect them. Hence, active exploration forms a vital and unavoidable element; (2) UAVs can exchange data with others located within a set distance using decentralised methods, enabling the sharing of newly located tasks; and (3) UAVs have the option to estimate other UAVs' probable intentions regarding contested tasks and shift priority toward those with fewer competitors.

The routing problem statement

We begin with the definition of the ideal cooperative routing challenge for multiple UAVs, assuming complete visibility, i.e., the search component has already been completed, and the full set of tasks is known in advance. However, our actual problem deviates in several fundamental aspects: (1) Only partial knowledge of task positions is available; (2) Agents function without central coordination; (3) The situation evolves continuously because agents are in motion and tasks appear or are finished as time passes; (4) Agents face no obligation to originate from or return to a fixed base location.

First, let indexes i and j indicate tasks drawn from the complete set T, numbered from 1 to N. Let A represent the set of agents numbered from 1 to M, and let the matrix $Cija$ signify the travel cost for agent a moving from task i to task j. In addition, we introduce the three-index binary decision variable:

© 2026 The Author(s). | Published by Agro Intelligence Research Press

$$x_{ija} = \begin{cases} 1 & \text{if agent } a \text{ visits task } j \text{ immediately after task } i, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The formulation then proceeds as follows:

$$\min_{x_a} \; Z \quad (2)$$

$$\text{s.t..} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij0} x_{ij0} \leq Z \quad (3)$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij1} x_{ij1} \leq Z \quad (4)$$

$$\vdots$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} C_{ijM} x_{ijM} \leq Z \quad (5)$$

$$\sum_{i=1}^{N} \sum_{a=1}^{M} x_{ija} = 1, \;\; \forall j \quad (6)$$

$$\sum_{i=1}^{N} x_{ipa} - \sum_{j=1}^{N} x_{pja} = 0, \;\; a \in A, p \in T \quad (7)$$

MATSP formulations generally aim to minimize either the overall distance covered or the total time required, both of which are identical when speeds remain constant (as in this case). For the challenge addressed in this work, the priority is finishing every task at the earliest possible moment, leading to minimisation of the highest individual agent completion time Z. Accordingly, the objective given in **Eq. 2** focuses on reducing this auxiliary variable Z. Constraints (3)–(5) guarantee that Z accurately reflects the longest single-agent distance, such that lowering Z achieves a balanced minimisation of the maximum individual distances. For agents operating at constant speeds, this maximum-distance minimisation directly translates into the time required to visit all tasks simultaneously, which later serves as the primary evaluation measure. Additional constraints include **Eq. 6**, which requires each task to be visited exactly once, and the flow conservation rules in **Eq. 7**, which stipulate that an agent leaving a task must have previously arrived there. Multiple techniques exist for handling sub-tours; the approach adopted here is the Miller–Tucker–Zemlin (MTZ) method, relying on the principle of ‘node potentials’ [16]. In this setup, u denotes supplementary integer variables, where ui relates to the ith

© 2026 The Author(s). | Published by Agro Intelligence Research Press

task. These variables assign ordering values to tasks and require that the sequence of visited locations follows increasing u values. This restriction ensures that each agent follows at most one continuous route.

The formulation presented constitutes the ideal centralised optimisation under conditions of total task knowledge and full coordination. Because the MATSP belongs to the NP-hard class of problems [17], its computational demands grow rapidly with additional variables or constraints. Increasing either the agent count or the task count can therefore lead to instances that require impractical computation time. Direct solving methods are thus often unworkable and particularly ill-adapted for decentralised operation or environments with incomplete information. To accommodate the demands of our decentralised, limited-information searching and routing scenario, several significant adaptations are introduced.

Partial-observability of task locations

Because this paper centres on a core search component, the locations of tasks remain unknown at the outset. Restrictions stemming from UAV sensing capabilities and communication limitations prevent agents from comprehending the full environment condition at once. In addition, the simultaneous and independent operation of agents renders the entire setup Partially Observable (PO). Nevertheless, a progressive understanding of current environmental conditions builds gradually through exploration, direct visual detection, and exchanges among agents. Of key importance, each agent maintains a unique perspective on the overall state, resulting in its private collection of tasks $Ta \subseteq T$.

Decentralised agents

Individual agent a navigates the surroundings and integrates every newly identified task into its dedicated task list Ta. Agents could also distribute these task lists to nearby agents. Still, since unrestricted information exchange is not consistently achievable and every agent deals with partial observability, a unified centralised answer to the setup in **Eqs. (2)–(7)** cannot practically be established. Agents therefore proceed independently, each tackling a TSP using solely its personal task list Ta. This transforms the overall challenge into a collection of possibly intersecting Single-Agent Travelling Salesman Problems (SATSPs). Conveniently, addressing a SATSP proves almost invariably far less complex than handling a MATSP. On top of that, the task list being merely the incompletely perceived portion $Ta \subseteq T$ renders each SATSP simpler yet, thanks to fewer tasks involved. These combined elements enable the use of basic heuristic techniques to resolve each agent's SATSP.

Non-stationarity

The combination of partial observability, decentralised agent operations, ongoing agent movements, and continuous task completion over time makes the problem non-stationary. An arrangement viewed as ideal during one moment can lose that status shortly afterward. Hence, instead of establishing one fixed overall solution for all agents at time t=0, the process demands adaptive responses through re-solving during each time-step to integrate recent details. This also calls for repeated updates to C. As a result, the evolving impact on C, along with agents' decentralised setup and their separate awareness of task positions, makes a unified three-dimensional cost matrix C unsuitable and unworkable. Every agent therefore keeps its personal two-dimensional cost matrix Cij, which can alter in size each time-step due to shifting tasks and requires refreshing to mirror the agent's latest location.

No fixed depot

Given the problem's fluid nature, the typical TSP rule requiring agents to originate from or return to a depot (stationary point) is relaxed. This is achieved by treating each agent's current position as a dummy task that serves as its private depot. Paired with this adjustment comes an asymmetric modification to the cost matrix Cij, where the cost Caj receives standard calculation from the agent's immediate spot (its dummy task) toward all remaining tasks, but the cost for tour completion Cja (meaning movement from an end task back to the dummy agent-task) equals zero.

Heuristic solution process

© 2026 The Author(s). | Published by Agro Intelligence Research Press

With no insistence on a fully optimal global answer to the setup in **Eqs. (2)–(7)**, multiple heuristic strategies can be utilised to deliver reliable solutions both speedily and dependably. Moreover, the problem's changing properties require route optimisation at every time step to accommodate newly arrived details (such as newly located tasks). Consequently, swift heuristic procedures are utilised to derive close approximations for the SATSP. During each time-step, agent a assembles its cost matrix Cij and randomly picks one from two heuristic methods (labelled Hexhaustive and H2opt) aimed at refining its present route.

The initial heuristic Hexhaustive applies a complete brute-force examination and appears in **Algorithm 1**. Agent a holds a cost matrix Cij together with an active route r. This route r forms an ordered sequence of tasks for visitation and offers another way to express the matrix xij shown in **Eq. 1**. The active route may derive from an earlier strong outcome or a random start. Route cost calculation uses this function:

$$\textbf{evalRoute}\,(r,C) = \sum_{i=1}^{L_r-1} C\,[r\,[i]\,,r\,[i+1]] \quad (8)$$

Where Lr stands for the length of route r. This matches the left side of **Eq. 3**, though applied to a sequenced list of tasks. Heuristic Hexhaustive reviews costs across all achievable route-order arrangements and selects the one $r*$ showing the smallest cost $c*$.

- **Input:** C, r, perm_limit
- $c^*, r^* \leftarrow$ **evalRoute**$(r, C), r$ ▷ Current best cost and route
- R_perms$\leftarrow$**permGenerator**$(r$,perm_limit) ▷ Generate all permutations
- **for** $r' \in$ R_perms **do**
- $c' \leftarrow$ **evalRoute**(r, c') ▷ Get cost of potential route
- **if** $c' < c^*$ **then**
- $c^*, r^* \leftarrow c', r'$ ▷ Update best cost and route
- **end if**
- **end for**
- **return** r^*, c^*

Algorithm 1. Hexhaustive solver for the SATSP.

Possible arrangements increase exponentially alongside route size, or more exactly by the factorial of route length $Lr!$. An assessment boundary termed perm_limit is therefore applied. The generator function permGenerator, shown in **Algorithm 2**, randomly picks and delivers one feasible route arrangement. When the number of arrangements exceeds perm_limit, it is not possible to test every option. Notably, permGenerator repeatedly generates a new random set of arrangements to prevent any bias toward a fixed sequence. A permutation index i is first set to point to the ith possible arrangement among all possible arrangements. Random draws from the span of permutation indexes then form a sample group $\{X1,\cdots,Xperm_limit\}$ equal in size to perm_limit. This group helps produce a selection of arrangements. Algorithm 2 relies on divmod (also known as Euclidean division) for obtaining quotient and remainder before pairing them with a version of the Fisher-Yates shuffle [18] to yield the indexed arrangements.

© 2026 The Author(s). | Published by Agro Intelligence Research Press

```
Input: r, perm_limit
Lr←length(r)
no_perms←min(Lr!,perm_limit)                    ▷ Get total allowable permutations
pIndexes←{X1,⋯,Xperm_limit}~U([1,⋯,no_perms])   ▷ Sample indexes
R_perms←{}                                      ▷ Keep a running set of permutations
for p in pIndexes do
  for i←1 to Lr−1 do
    p,j←divmod(p,Lr−i)                          ▷ Get quotient and remainder
    r[i],r[i+j]←r[i+j],r[i]                     ▷ Swap the elements
    R_perms←R_permsU{r}                         ▷ Append to set of permutations
  end for
end for
return R_perms
```

Algorithm 2. permGenerator - Limited permutation generator.

The following heuristic function H2opt stems from the familiar TSP heuristic, the two-opt swap, initially proposed by Croes [19] and employed effectively as a direct yet potent approach for various optimisation challenges [20-23]. Basic two-opt seeks to pick two neighbouring nodes [⋯,,+1,⋯] inside a route and reverse them to [⋯,vi+1,vi,⋯] hoping that crossed paths might straighten. Procedure H2opt is laid out in Algorithm 3 and follows this swapping concept by cycling through each node in route r to perform the swap. Our version also includes a second neighbourhood loop to check extended swap distances. Specifically, for neighbourhood size Nr, all elements from i to i+k undergo swapping where k∈(1,⋯,Nr) instead of merely i and i+1. The paired loops roughly echo breadth-first and depth-first search styles. An assessment boundary (eval_limit) applies here as well to keep computation durations in check.

```
Input: C, r, Nr, eval_limit
c*,r*←evalRoute(r,C),r                          ▷ Current best cost and route
Lr←length(r)
evals←0
for i←1 to Lr−1 do
  for j←i+1 to min(Lr,i+1+Nr) do
    if j−i≠1 then
      r′←[r[0:i−1]+reverse(r[i:j−1])+r[j:Lr]     ▷ Swap the order
      c′←evalRoute(r′,C)
      evals←evals+1
    end if
    if c′<c* then
      c*,r*←c′,r′                               ▷ Update best cost and route
    end if
    if evals>eval_limit then                    ▷ Check if eval limit reached
      Break
    end if
  end for
end for
return r*,c*
```

Algorithm 3. Two-Opt Heuristic solver H2opt for the SATSP.

© 2026 The Author(s). | Published by Agro Intelligence Research Press

This search task's specific features cause agents to continuously roam while refreshing task lists. Therefore, optimal choices can evolve swiftly. In these conditions, adequate solutions often suffice and require less processing. The two heuristics target somewhat separate zones within the solution space. Hexhaustive functions as a fully random brute-force tactic that hops across the global space seeking upgrades, whereas H2opt enhances a present route via nearby tweaks. Their joint application exceeds the requirements for strong individual-agent solutions while remaining efficient for per-time-step execution (growing linearly with the number of agents). Besides, each agent stores its current task sequence and route, allowing ongoing enhancements in later time steps. Significantly, these techniques avoid complete restarts from zero, unlike many other optimisation methods. Fresh tasks are integrated into the route at a suitable spot; here, they are appended to the end of the route. Retaining the core route structure presumes the new task will at least follow the existing best sequence. Placement at the end, moreover, provides additional time steps for heuristics to adjust and optimise the visitation order.

UAV Agent Behaviours

A foundational 'base' agent behaviour is established first, acting as the origin for deriving additional agent behaviours. This behaviour is grounded in the Agent-Based Modelling and Simulation framework [24, 25], positioning agents as independent, self-managing, and complete units. Agents reside inside the environment, engage with it plus fellow agents, and respond to influences. Simulation places multiple such agents into the setting and progresses through several discrete time steps or until a target outcome is reached. Within every time-step, each UAV agent carries out these steps:

- Sense: Acquire data from surroundings using sensors and from fellow agents using communication
- Plan: Analyse received data plus retained memory to select a route for pursuit and/or travel heading
- Act: Carry out a motion toward the selected heading and possibly share data with others

As depicted in this paper and in **Figure 1**, UAVs detect task positions visually. When a task enters a set within a nearby distance called the 'vision radius' Rvision in that time-step, detection occurs. UAVs also feature a 'communication radius' Rcomms defining the circular span for sending signals to other agents and picking up replies. From **Figure 1**, it is evident that UAV1's selection of task order might be affected by details of extra tasks beyond its immediate field of view. Hence, an agent's behaviour rests on three main features:

© 2026 The Author(s). | Published by Agro Intelligence Research Press

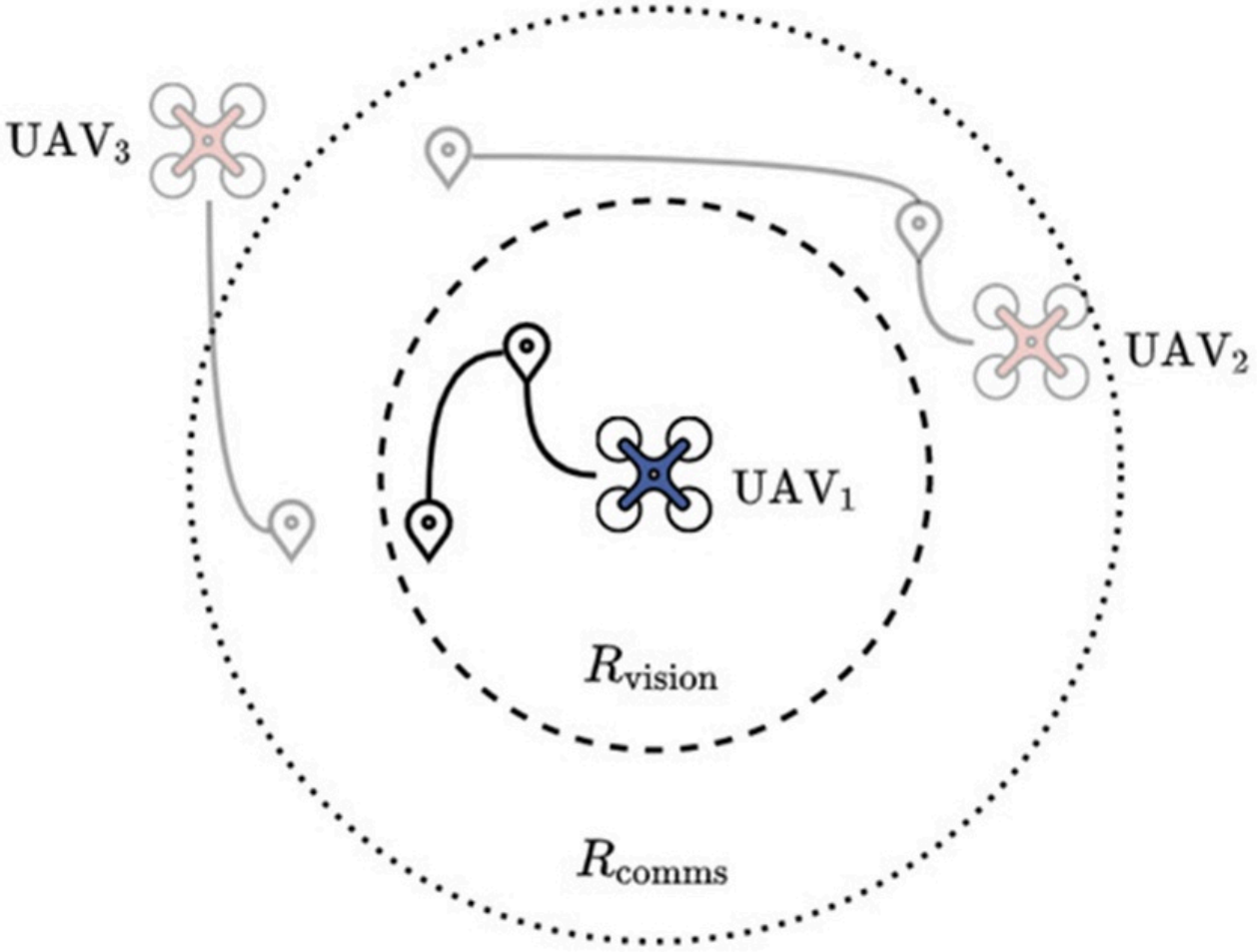


Figure 1. Diagram of UAV1's communication radius Rcomms and vision radius Rvision for finding tasks and communicating with other agents (UAV2 and UAV3).

- Speak: Content an agent sends outward to others.
- Listen: The manner in which an agent handles incoming communications.
- Model: The approach an agent takes to anticipate other agents' actions.

Considering these features, a driving question for this paper is whether team-wide results can improve when agents mutually distribute known tasks. Though greater environmental awareness might seem certain to improve performance, this paper shows that this is not always the case.

Activating or deactivating solely two features (listening and modelling) on the base agent produces four unique behaviours. Summarised in **Table 1**, they receive the names Solo, Greedy, Solo+, and Greedy+. A deeper discussion of the speak and listen features, covering details on substance, schedule, and process, occurs in next sections.

Table 1. Three agent behaviour types from three base modifications.

Strategy type	Communication behavior	Reception behavior	Decision model

© 2026 The Author(s). | Published by Agro Intelligence Research Press

Solo	Sends out every task to others	Does not take in any received tasks	No model applied
Greedy	Sends out every task to others	Accepts every task it receives	No model applied
Solo+	Sends out every task to others	Does not take in any received tasks	Advances the next competing task
Greedy+	Sends out every task to others	Accepts every task it receives	Advances the next competing task

Modelling other agents' intent

Modelling the intentions of fellow agents means attempting to foresee and interpret their likely next actions to refine one's personal decisions. For this study, the modelling adopts an elementary approach grounded in one central query: Might a different UAV arrive at a given task earlier? **Figure 2** demonstrates this situation with two agents — Blue positioned left and Red positioned right — contending over tasks A, B and C. Treating ourselves as the Blue Agent, we pursue a path visiting A followed by B then C. While unaware of Red's exact goals, we suppose in this illustration that Red pursues the same three tasks using the same sequence. In the initial situation, Blue maintains its plan unchanged, enabling Red to secure every task in advance. In the revised situation, Blue notices Red's location (as it falls inside the Rvision range), determines Red stands nearer to task A, and through estimating Red's likely objective toward A, realises it cannot reach the spot sooner. Blue therefore defers task A, pushing it to the final position in its sequence and advancing instead toward the subsequent unclaimed task (B in this instance). Hence, a UAV with this modelling characteristic can utilise positional data from nearby agents to shape its path strategy more intelligently, ideally avoiding direct clashes over identical targets and thereby accomplishing alternative targets with greater effectiveness.

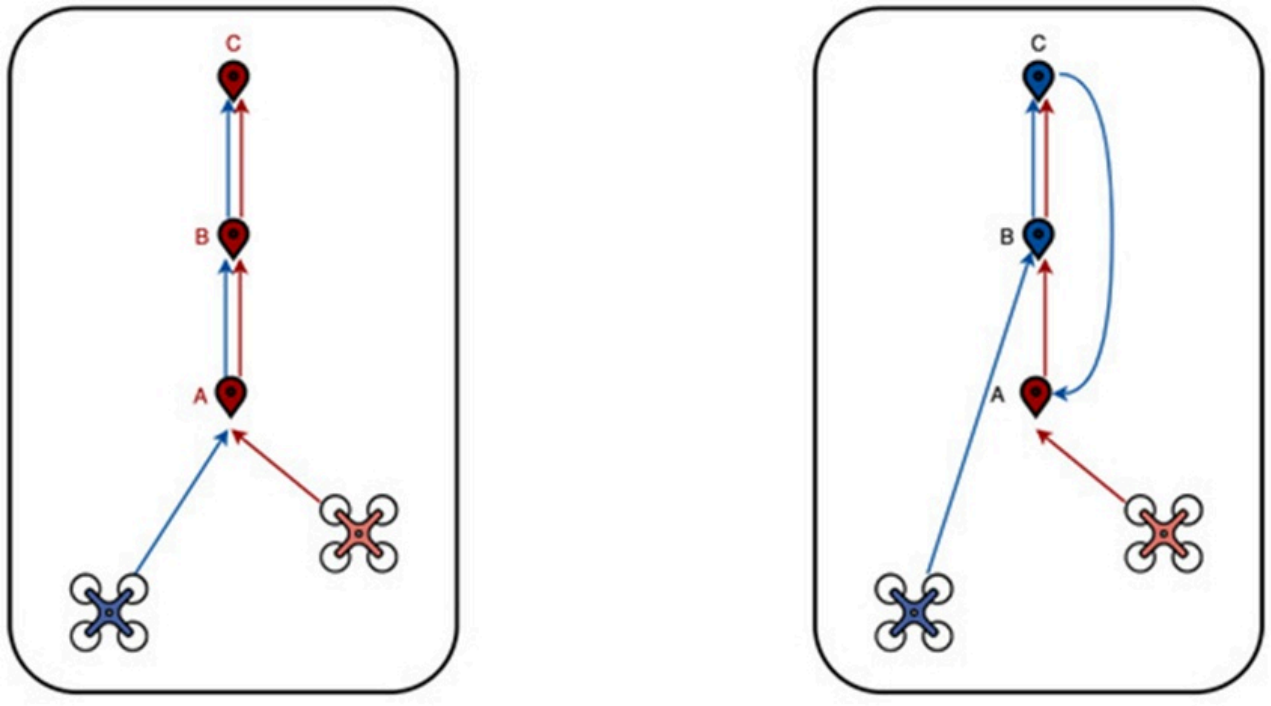


Figure 2. Two agents competing for three tasks. Blue agent (left) might change the order in which it completes its tasks based on the assumption that Red agent (right) might reach a task first.

© 2026 The Author(s). | Published by Agro Intelligence Research Press

Multi-Agent System Simulation Environment

Examining the influence of these behaviours, along with their adjustable settings, on outcomes requires the creation of a simulation platform. Core prerequisites for the investigation were (1) full Python implementation; (2) inclusion of a compact Graphical User Interface enabling visual demos and parameter tweaking; (3) capacity to execute groups of repeated runs and output performance metrics. Selection fell on the Python-based Agent-Based Modelling Simulator MESA [26], which provides essential ABMS foundations and a user interface. Supplementary modifications were applied to the standard MESA codebase to faithfully capture distinctive features of our MSSRP.

The primary structural elements in our arrangement include Agents, Spaces, and Schedules. Agents are active participants endowed with autonomy — performing actions such as relocation, interaction, and state updates. Space refers to the surrounding domain containing the agents, which can take the form of networks, grids, or open fields. Agents connect to a scheduler that triggers their individual step functions and dictates the activation sequence. Scheduler operation may follow ordered, simultaneous or — as implemented here — randomised patterns. Consequently, a simulation positions Agents within their respective Space(s), then, at successive time-steps, activates the scheduler(s) to execute all pertinent Agents' step functions, repeating until a predefined end condition is met.

The environment

The simulated world integrates two separate spaces. The first consists of a two-dimensional continuous area inhabited by UAV agents. The second comprises a hexagonal grid populated by terrain cells. These hex cells fall into four distinct categories: sea (blue), shore (yellow), land (green), and hill (dark green). Procedural rules generate the tile configurations, producing diverse yet coherent map patterns (**Figure 3**). A randomly chosen land tile acquires the 'base' designation (rendered in red) to indicate UAV launch coordinates. Significantly, these generated layouts avoid purely stochastic task distribution by confining tasks exclusively to land-type tiles (shore, land, hill), thereby replacing uniform randomness with clustered patterns. In the current context, terrain primarily provides a visual indication of feasible task sites. Features such as island quantities, land-to-sea proportions, and base placement can clearly advantage particular agent strategies and mutual influences. It should be noted that the hexagonal grid lies within the continuous domain, ensuring that every process—UAV navigation and task positioning alike—proceeds continuously.

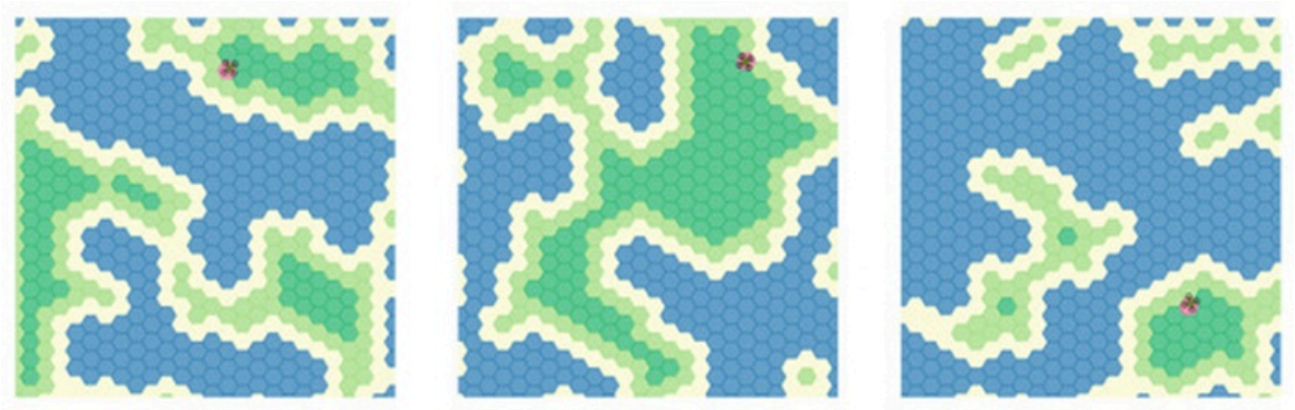


Figure 3. Three examples of procedurally generated maps of hexagonal terrain tiles (blue, yellow, green, dark green), a base spawn location (red), and a quad-copter UAV at the base.

Task agents

Tasks operate as elementary Agents introduced once into the environment, where they remain immobile. They monitor various characteristics, such as visitation frequency and current occupancy. They further evaluate whether fulfilment has occurred through a UAV Agent's visit. Tasks here emerge solely upon shore, land, or hill hexes inside environment limits. Generation involves repeated random position trials until an eligible tile site is identified.

© 2026 The Author(s). | Published by Agro Intelligence Research Press

UAV agents

UAV Agents, along with the associated behaviours, represent the central focus of both this paper and the accompanying simulations. Like standard ABMS Agents, they activate their step procedure in response to the scheduler signal. Step procedures for Tasks and Terrain cells remain elementary, chiefly maintaining records of occupancy and completion indicators. Conversely, the UAV agent's step procedure incorporates navigation, path determination, and data sharing. The UAV agent's step procedure proceeds according to this sequence:

- Get Visible
- Get Communicable
- Receive Messages (Listen)
- Optimise Route
- Send Messages (Speak)
- Move

Get visible

Identification of perceptible agents in the surroundings is performed here, encompassing both tasks and UAVs. Beginning with the entire task collection T, the Get Visible operation isolates the subset $Tvis \in T$ applicable to a specific agent a lying within the prescribed radial span Rvision measured from the agent's current coordinates. Precisely:

$$\mathrm{Tvis}_a = \{t_l \in T,\ \text{for}\ \ i \in (1 \cdots N) \mid \mathbf{dist}\,(A_a, t_l) \leq R_{\mathrm{vision}}\}. \quad (9)$$

The function dist computes the Euclidean distance (L2 Norm) between any pair of agent coordinates. From the complete agent collection A, the Get Visible operation similarly isolates the subset $Avis \in A$ for agent a residing inside the prescribed radial span Rvision from its present coordinates:

$$\mathrm{Avis}_a = \{A_b \in A,\ \text{for}\ \ b \in (1 \cdots M)\ a \neq b \mid \mathbf{dist}\,(A_a, A_b) \leq R_{\mathrm{vision}}\}. \quad (10)$$

Get communicable

Following the same logic, from the entire agent collection A, the Get Communicable operation isolates the subset $Acomms \in A$ for agent a residing inside the prescribed radius Rcomms. Execution utilises this equation:

$$\mathrm{Acomms}_a = \{A_b \in A,\ \text{for}\ \ b \in (1 \cdots M)\ a \neq b \mid \mathbf{dist}\,(A_a, A_b) \leq R_{\mathrm{comms}}\}. \quad (11)$$

Configurations displaying assorted pairings of Rvision and Rcomms values among agents could merit attention. Nevertheless, this paper treats all UAVs as homogeneous, employing uniform Rvision and Rcomms values.

Receive messages (Listen)

© 2026 The Author(s). | Published by Agro Intelligence Research Press

Agent a may accept transmissions dispatched in the preceding time-step from every fellow agent inside its Acomms$a$ collection. Simulator oversight ensures transmissions endure for merely one time-step and reach destinations exclusively when distance criteria meet Rcomms thresholds and behaviour rules permit. Per **Table 1**, behaviour classification may direct an agent to incorporate or discard incoming data. Messages here contain solely task lists accompanied by coordinates. Upon valid reception, the agent appends unfamiliar tasks to the conclusion of its existing task collection.

Optimise route

To collect active tasks, the agent must establish an optimal visitation sequence. This sequence inherently creates a successive path the UAV can pursue. Maximising path effectiveness involves applying the two TSP heuristics.

Incorporation of fellow-agent intent modelling also occurs at this juncture. Execution draws upon the proximate visible agent collection Avis$a$ defined in **Eq. 10** along with the technique outlined in previous sections. Concretely, should the imminent task lie nearer to any agent within Avis$a$ we forecast prior visitation by that agent and consequently shift the task to the conclusion of our active path.

Send messages (Speak)

Agents gain the capability to dispatch data externally (without certainty of reception). Transmitted content here comprises the full task list recognised by the sending agent. Simulator controls again restrict delivery to qualified recipients only.

Move

Agents terminate their step procedure via locomotion, projecting their motion dynamics forward by a single interval dt. At each time step, the UAV agent determines two parameters: the target direction v (a two-dimensional velocity vector) and the target speed s. These feed into the agent's control module, which computes the realised velocity and speed. The control module observes physical boundaries covering rotation limits, peak acceleration or deceleration, and velocity ceiling. Resulting directives execute, displacing the agent across the space.

Having an active task collection (and the associated path from Optimise Route) directs the UAV toward the next target. Without tasks, the agent adopts exploration mode. Exploration utilises random walk generation through alteration of the current direction vi via random deviation $\mathcal{V}$,

$$v'_{t+1} = v_t + \mathcal{U} \quad \text{where} \quad \mathcal{U} \sim \mathcal{U}_2(a, b) \tag{12}$$

Where $\mathcal{U}2$ signifies a two-dimensional uniform distribution. Simulation settings confine the bound from a = −0.125 to b = 0.125. Across all scenarios, the desired speed s initiates at and remains locked to the uppermost allowable limit, set here at 5 m/s.

Simulation assumptions

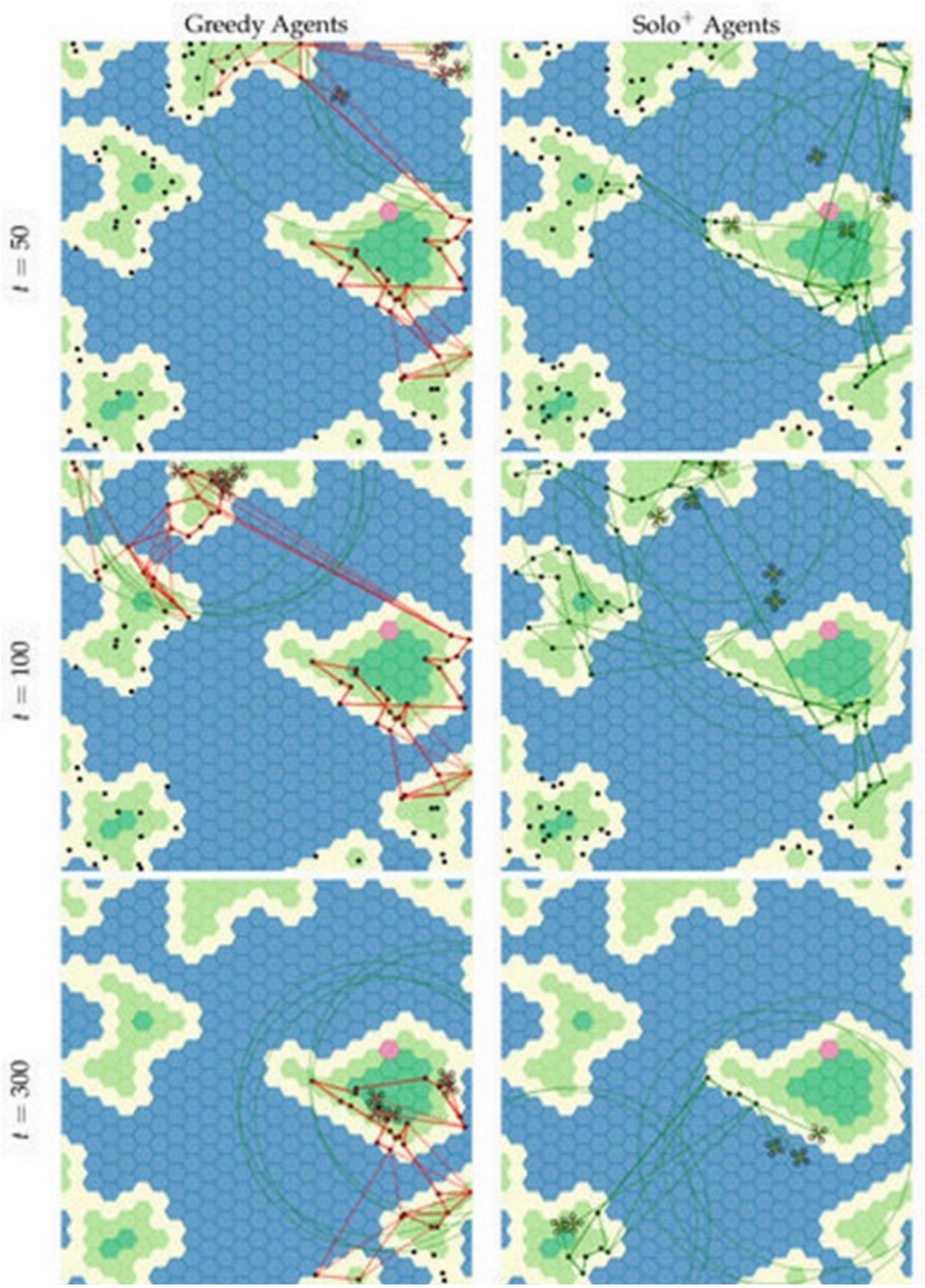
Although these choices do not represent restrictions within the problem definition or the simulation platform itself, several assumptions were adopted for the current paper and the outcomes presented. First, every UAV begins from one shared base position (the red hex shown in **Figure 4**), yet the precise spot inside that hex varies randomly across UAVs. Allowing varied starting positions could yield advantages in certain situations, yet examining that factor would introduce extra variables not needed here. No limitations apply to fuel usage, as each UAV launches with unlimited fuel reserves and expends none during the entire run. Incorporating fuel considerations would likely elevate the importance of base positioning (the refuelling site) as a major factor affecting outcomes.

Furthermore, UAVs must remain within the simulation boundaries, with the ABMS Space responsible for redirecting them inward upon edge contact (visualised as geo-fencing). Since tasks appear exclusively inside the boundaries, this

© 2026 The Author(s). | Published by Agro Intelligence Research Press

boundary handling matters only during random walk phases when no tasks are assigned. Potential UAV collisions receive no attention, and no separation rules apply. Lastly, UAVs are not obligated to return to base after each task, and the run ends upon full task completion. The precise influence these assumptions exert on findings and interpretations remains unexamined here, yet they open avenues for valuable future studies.

© 2026 The Author(s). | Published by Agro Intelligence Research Press



© 2026 The Author(s). | Published by Agro Intelligence Research Press



*R*comms of 50 m. With the left column using all Greedy behaviours and the right column using all Solo+ behaviours. Black dots indicate task locations, dashed lines are each UAVs intended path (many overlap), and the green circle indicates *R*vision.

Example of single simulation run

An illustrative single run now displays the described methods and simulation steps. This run compares two behaviour categories: Greedy and Solo+. Both cases involve 5 UAVs sharing identical behaviour and settings. Every UAV features a visual range *R*vision of 250 m (equalling one third of the 750 m by 750 m environment dimensions). Communication range *R*comms equals 50 m (one-fifteenth of the environment dimensions), restricting exchanges to UAVs positioned very near each other.

Progression across each run is shown in snapshots in **Figure 4**, captured at time steps 50, 100, and 300 (displayed row-wise). Black dots mark task positions while dotted lines show each UAV's present planned sequence for visiting them. Because numerous UAVs have either observed or learned identical tasks, their planned sequences frequently coincide (multiple overlapping dotted lines may visually merge into a single continuous line). Such collective awareness of the environment, absent deliberate coordination, often leads to groups of UAVs converging on the same tasks concurrently. This leads to suboptimal utilisation of the available UAV fleet. The grouping stands out more prominently among Greedy Agents in **Figure 4** (left column), yielding a completion time of 483 time steps versus 325 time steps for Solo+ (right column). This pattern of grouping and rivalry over shared tasks is examined in greater depth in next section and constitutes a primary driver of reduced effectiveness.

Results and Discussion

Parameter space exploration now proceeds through batches of repeated simulations. One individual trial receives definition from the settings listed in **Table 2**. The upper rows of the table contain parameters held constant across all trials, while the lower rows list the adjustable parameters that are systematically varied. This Section investigates how selections among these adjustable parameters shape collective outcomes.

Table 2. Fixed and variable Trial Parameters used within the simulation.

Parameter	Description	Value
Max steps	Maximum number of simulation time steps allowed	1000
*T*0	Initial number of tasks generated at the start	100
T+	Tasks generated over time during the run	0
Environment size	Dimensions of the simulated world (width × height)	750 m × 750 m
Hex Size	Distance from the center to the vertex of each hex cell	25 m
N	Total number of UAV agents	[2, 5, 10, 20]
*R*comms	Communication range radius (meters)	[50, 250, 1500]
*R*vision	Vision/sensing radius (meters)	[25, 50, 100, 250, 500, 1500]
Behaviour	Type of agent decision strategy	[Solo, Greedy, Solo+, Greedy+]

A complete configuration combines the 5 unchanging and 4 adjustable parameters from **Table 2** and undergoes 25 repetitions against a fixed collection of 25 trials. Within each trial for a given configuration, all agents share matching

© 2026 The Author(s). | Published by Agro Intelligence Research Press

settings (identical behaviour type along with uniform Rcomms and Rvision values) and therefore operate homogeneously. The full set comprises 25 trials for each of the 288 (4×3×6×4) distinct configurations drawn from **Table 2**. Each of these 25 trials is initialised with a matching seed key, ensuring each configuration encounters the same consistent trial set. The terrain map used throughout these outcomes is shown in **Figure 4** and loaded at simulation start. Each of the 25 seed keys initialises a pseudo-random number generator (RNG). This RNG determines the starting positions for tasks and UAV agents; its seeded nature ensures reproducibility, enabling 25 distinct yet repeatable trials for evaluation across all configurations. All remaining stochastic elements in the simulation draw from separate, unique RNG streams initialised via the system clock.

With this framework established, the outcomes from our batch simulations appear across the four-dimensional sweep covering Rcomms, Rvision, UAV count N, and Behaviour. Per the definition in in previous sections, the goal focuses on optimising the objective in **Eq. 2**, which equates to minimising the total number of time steps taken until all tasks finish. Outcomes are displayed in **Figure 5** and **Figure 6**, containing identical information yet oriented differently to highlight (i) Rvision **(Figure 5)**; (ii) Rcomms **(Figure 6)**; (iii) UAV count (rows); or (iv) Behaviour (lines). Solid lines for each behaviour/colour show the mean across 25 runs, while shaded regions indicate result variability (standard deviation).

© 2026 The Author(s). | Published by Agro Intelligence Research Press

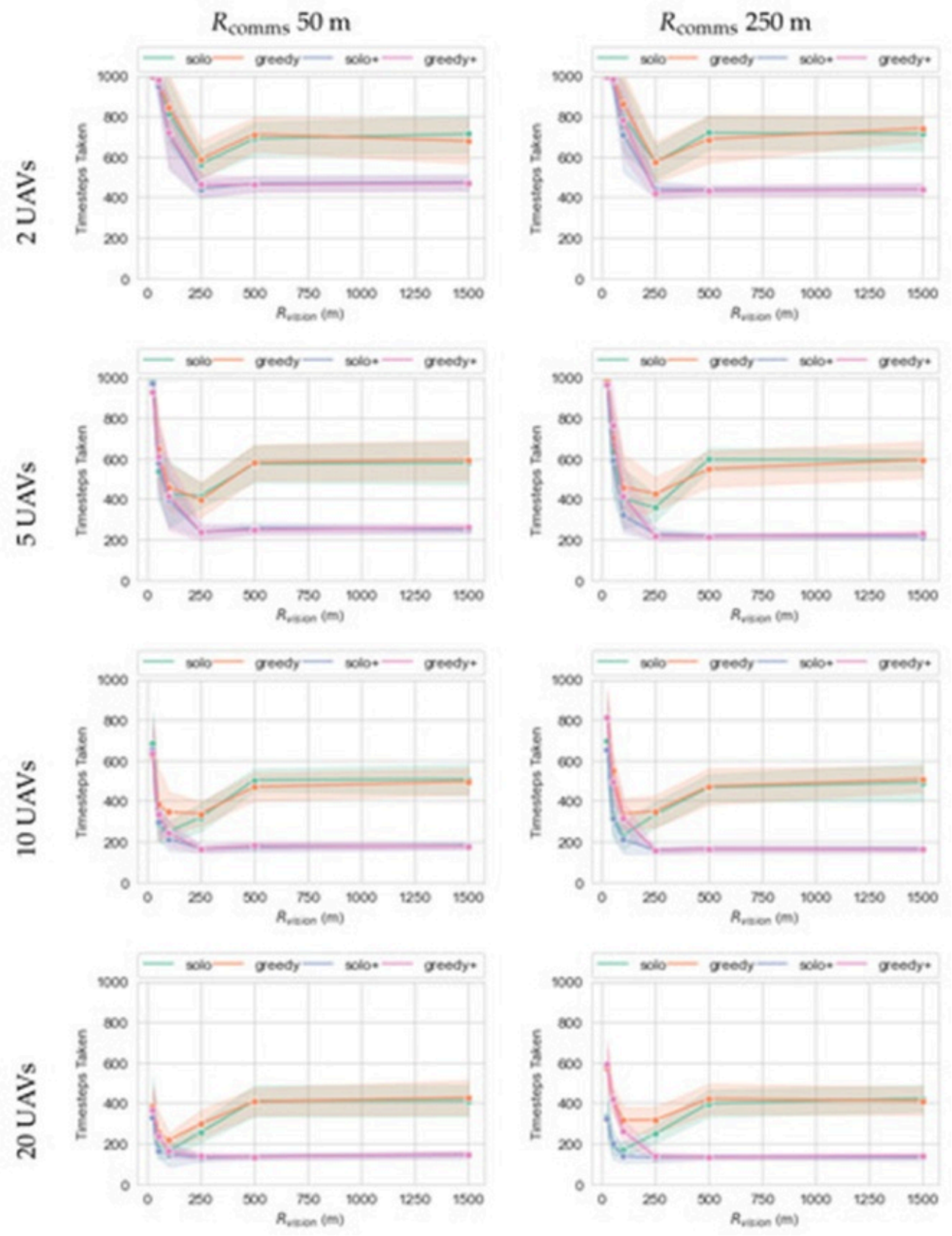
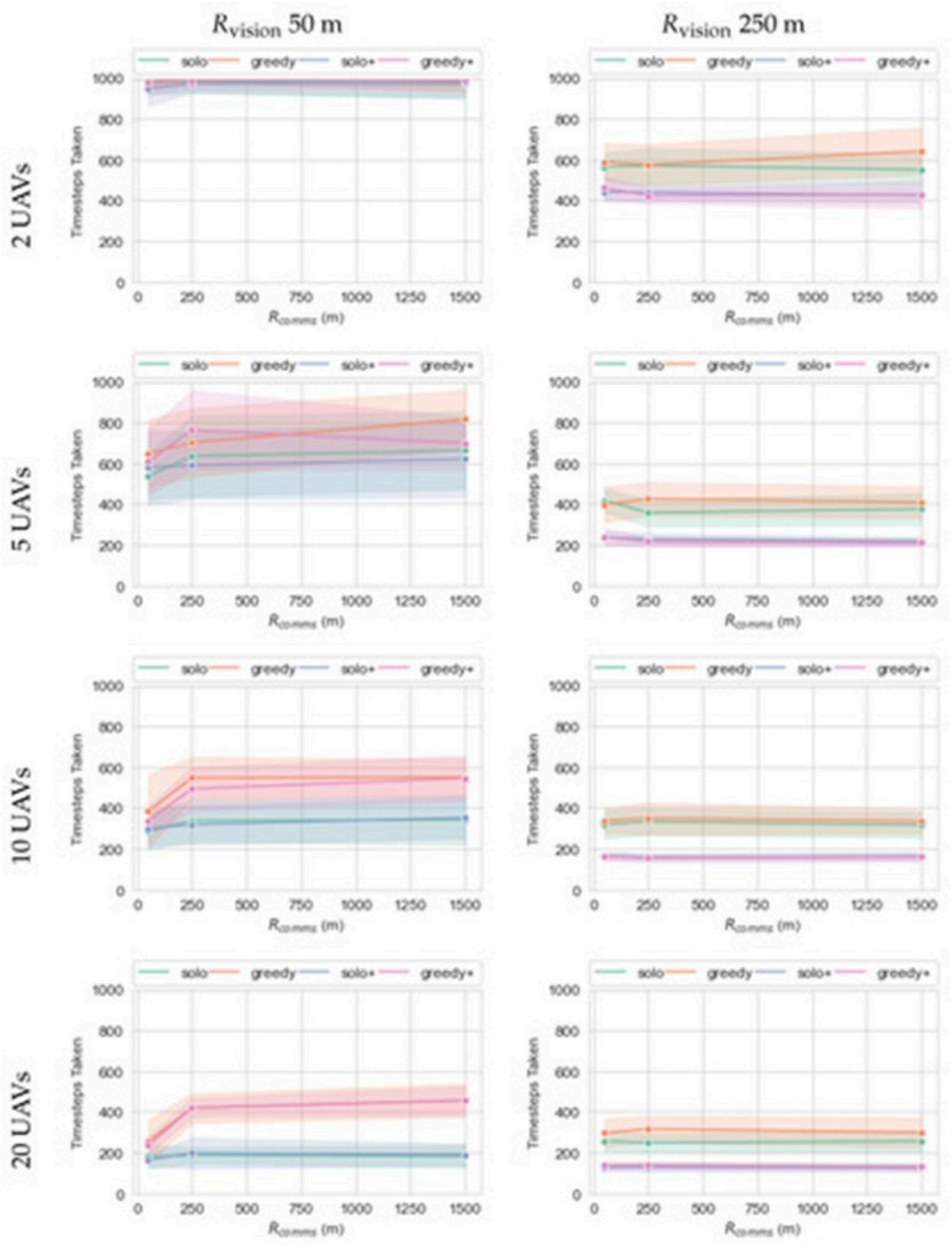


Figure 5. A sweep over Rvision values for increasing (top to bottom) numbers of UAVs for two fixed Rcomms values of 50 and 250 m.

© 2026 The Author(s). | Published by Agro Intelligence Research Press



© 2026 The Author(s). | Published by Agro Intelligence Research Press

Figure 6. A sweep over Rcomms values for increasing (top to bottom) numbers of UAVs for two fixed Rvision values of 50 and 250 m.

Initial examination of UAV count effects in **Figures 5 and 6** reveals a distinct pattern. Increasing the number of UAVs from 2 to 20 (top to bottom rows) improves outcomes across all behaviour categories. Nevertheless, this improvement occurs inefficiently: expanding from 10 to 20 agents fails to halve the required time steps. Such scaling inefficiency causes the aggregate time-steps across all UAVs (M×Z) to rise more rapidly than Z falls. If efficiency is ranked as a secondary priority, trade-offs would arise between optimal UAV deployment numbers and overall returns.

Clear patterns also emerge concerning restrictions on agent vision range Rvision and their consequences for collective effectiveness. Constraining Rvision directly curtails a UAV's capacity to discover fresh tasks during exploration, causing markedly weak outcomes in cases with especially small Rvision (< 100 m). Conversely, beyond a threshold (> 1000 m), a UAV can continuously monitor every task regardless of position, rendering the search aspect largely irrelevant. **Figure 5** consistently displays low Rvision producing weak results, an optimum near 250 m, and diminishing further gains at elevated values. Notably for solo and greedy behaviours (those lacking modelling), excessive Rvision harms effectiveness. Increased visibility increases the likelihood that UAVs will pursue identical tasks and travel in clusters rather than disperse. Beyond direct task rivalry, expanded task lists likely diminish the performance of heuristic routing methods from previous section, yielding poorer sequences.

The pattern shifts somewhat when altering Rcomms. Recall that the sole distinction between solo and greedy lies in greedy agents incorporating heard tasks into their lists while solo agents disregard incoming messages. Consequently, communication exerts no influence on solo or solo+ agents. As visible in **Figure 5**, shifting Rcomms from 50 m (left panels) to 250 m (right panels) produces limited overall impact. Closer inspection of **Figure 6** shows largely flat responses to Rcomms changes in most cases. The notable exception involves greedy and greedy+ agents paired with Rvision of 50 m (left panels), where expanded communication range produces negative effects. This outcome aligns with findings from elevated Rvision and supplies further support that merely acquiring awareness of additional tasks proves insufficient for better performance. Portions of this outcome likely stem from inherent problem traits, given the spatial character of searching, communication constraints, and uniform sensing capabilities across agents. A task's proximity to an agent strongly correlates with the probability of visitation once awareness exists. Informing distant agents about nearby tasks therefore offers minimal advantage and may only complicate other agents' SATSP difficulties.

Importantly, we can easily observe the effect of adding modelling capability to the UAVs, i.e., Solo+ and Greedy+, on performance.

It becomes evident from **Figure 5** that the absence of modelling causes rising Rvision to impose a substantial negative impact on outcomes. This stems from UAVs acquiring extensive overlapping task lists and facing a higher risk of rivalry over those tasks, which reduces overall efficiency. In essence, the modelling addition offsets the harmful consequences of excessive environmental awareness (that is, increased exposure to tasks). Notably, runs involving just 2 UAVs equipped with modelling and Rvision exceeding 250 m can surpass the results achieved by 10 UAVs lacking modelling. Hence, within uniform UAV groups, intelligent handling of available environmental data is far more significant than merely expanding team size. Lastly, performance variability (represented by shaded regions) shows a marked reduction when modelling is included. In reality, no additional parameter exerts comparable influence on the result spread. Consequently, without any cooperation mechanism — here realised via modelling — system outcomes remain heavily dependent on luck.

The findings indicate that a principal source of weak multi-agent outcomes lies in inadequate workload allocation for task visitation. Solo-behaviour UAVs function in complete isolation, leaving deconfliction of rival actions dependent on stochastic factors or inherent environmental features that separate the agents. Greater awareness of the partially perceived surroundings, gained either via expanded sight for task detection or broader messaging to inform others of discoveries, heightens the tendency toward identical world representations. Uniform behaviour across agents translates

© 2026 The Author(s). | Published by Agro Intelligence Research Press

shared beliefs into parallel choices, thereby increasing the probability that UAVs converge on and contend for identical tasks. This spontaneous convergence phenomenon has been examined previously by the Authors in Ref. [5] in a multi-UAV monitoring scenario. That work particularly revealed that deliberate diversity, introduced through minor random perturbations in action selection or state perception, could counteract the issue.

Outcomes here identify two primary strategies for managing such convergence in choices and environmental understanding (comparable to [5]). First, partial observability may be an advantageous environmental feature, as divergent world perspectives among homogeneous agents enable indirect collaboration through limited knowledge. Second, and more encouragingly, agents need not rely exclusively on this limited knowledge; instead, they can estimate fellow UAVs' likely aims and avoid pursuing tasks others are expected to complete.

Conclusion

This paper examines the Multi-Agent Simultaneous Searching and Routing Problem, in which multiple homogeneous UAVs simultaneously locate unknown tasks within an environment and determine paths among them. Adjustments were applied to conventional global optimisation methods to accommodate characteristics such as decentralised operation and partial observability inherent in real settings. The research draws on efforts to reduce competitive behavior among UAVs lacking direct coordination. Specifically, the study illustrates that cooperative outcomes remain attainable via judicious parameter selection and straightforward behavioural adjustments. Activating or deactivating two particular attributes of a standard UAV design — reception of messages from others and basic estimation of fellow UAVs' aims — yielded four separate UAV behaviour variants. Deployment of a multi-agent simulation platform enabled testing and analysis across the UAV parameter space to identify ideal settings for effectively addressing the combined searching and routing challenge.

Findings from this paper establish that enhanced environmental awareness, acquired by detecting additional tasks or receiving notifications from other agents, can actually impair results when operating with multiple homogeneous agents. Indeed, partial observability can be leveraged through restricted vision and communication distances, enabling multiple UAVs to collaborate indirectly using incomplete information. Significantly, we demonstrate that dependence on such incompleteness becomes unnecessary when agents apply minimal estimation of other agents' intentions. Crucially, incorporation of this modelling approach ensures that greater world knowledge consistently yields superior performance.

Nomenclature

N	Total number of Tasks
T	Set of Tasks from 1 to N
a	Agent a
M	Total number of Agents
A	Set of UAV Agents from 1 to M
Ta	Subset of tasks for Agent a
$Cija$	Cost Matrix for agent a travelling between task i and j
$xija$	Binary decision variable task i, j, and agent a
r	Route
$r*$	Best route
c	Cost

© 2026 The Author(s). | Published by Agro Intelligence Research Press

$c*$	Best cost
LR	Length of route r
$\mathscr{U}$	Uniform distribution

Abbreviations

ABMS	Agent-Based Modelling Simulator
UAV	Unmanned Aerial Vehicle
TSP	Travelling Salesman Problem
MSSRP	Multi-Agent Simultaneous Searching and Routing Problem
MATSP	Multi-Agent TSP
SATSP	Single Agent TSP
PO	Partial Observability
RNG	Random Number Generator

Acknowledgements

None

Conflict of interest

None

Financial support

This work was funded and delivered in partnership between the Thales Group and the University of Bristol, and with the support of the UK Engineering and Physical Sciences Research Council Grant Award EP/R004757/1 entitled ‘Thales-Bristol Partnership in Hybrid Autonomous Systems Engineering (T-B PHASE)’.

Ethics statement

None

Received: 26 Aug 2024 Revised: 31 Oct 2024 Accepted: 15 Nov 2024
Published online: 25 November 2026

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and

distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at https://creativecommons.org/licenses/by-nc-sa/4.0/.

References

1. Ceccarelli N, Enright JJ, Frazzoli E. Micro UAV path planning for reconnaissance in wind. In: Proceedings of the American Control Conference; New York, NY, USA; 2007. p. 5310–5.

2. Nigam N. Dynamic replanning for multi-UAV persistent surveillance. In: Proceedings of the AIAA Guidance, Navigation, and Control (GNC) Conference; Boston, MA, USA; 2013. p. 1–20.

3. Nigam N, Kroo I. Persistent surveillance using multiple unmanned air vehicles. In: Proceedings of the IEEE Aerospace Conference; Big Sky, MT, USA; 2008. p. 1–15.

4. Li X, Savkin AV. Networked unmanned aerial vehicles for surveillance and monitoring: A survey. Future Internet. 2021;13:174.

5. Kent T, Richards A, Johnson A. Single-agent policies for the multi-agent persistent surveillance problem via artificial heterogeneity. In: Multi-Agent Systems and Agreement Technologies; Springer: Cham, Switzerland; 2020. p. 243–60.

6. Savkin AV, Huang H. Navigation of a UAV network for optimal surveillance of a group of ground targets moving along a road. IEEE Trans Intell Transp Syst. 2021;22:3891–6.

7. Savkin AV, Huang H. Asymptotically optimal path planning for ground surveillance by a team of UAVs. IEEE Syst J. 2021;1–4.

8. Kitano H, Tadokoro S, Noda I, Matsubara H, Takahashi T, Shinjou A, et al. RoboCup Rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In: Proceedings of the IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028); Tokyo, Japan; 1999; 6: 739–43.

9. Beck Z. Collaborative Search and Rescue by Autonomous Robots. Ph.D. Thesis, University of Southampton, Southampton, UK, 2016.

10. Liu Y, Nejat G, Vilela J. Learning to cooperate together: A semi-autonomous control architecture for multi-robot teams in urban search and rescue. In: Proceedings of the 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR); Linköping, Sweden; 2013.

11. Ma H, Tovey C, Sharon G, Kumar T, Koenig S. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In: Proceedings of the AAAI Conference on Artificial Intelligence; Phoenix, AZ, USA; 2016; 30.

12. Song BD, Park K, Kim J. Persistent UAV delivery logistics: MILP formulation and efficient heuristic. Comput Ind Eng. 2018;120:418–28.

13. Tin C. Robust Multi-UAV Planning in Dynamic and Uncertain Environments. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2004.

14. Sabo C, Kingston D, Cohen K. A formulation and heuristic approach to task allocation and routing of UAVs under limited communication. Unmanned Syst. 2014;2:1–17.

15. Amigoni F, Banfi J, Basilico N. Multirobot exploration of communication-restricted environments: A survey. IEEE Intell Syst. 2017;32:48–57.

16. Miller CE, Tucker AW, Zemlin RA. Integer programming formulation of traveling salesman problems. J ACM (JACM). 1960;7:326–9.

© 2026 The Author(s). | Published by Agro Intelligence Research Press

17. Liu H, Zhang P, Hu B, Moore P. A novel approach to task assignment in a cooperative multi-agent design system. Appl Intell. 2015;43:162–75.

18. Fisher RA, Yates F. Statistical Tables for Biological, Agricultural and Medical Research. Minneapolis, MN, USA: Hafner Publishing Company; 1953.

19. Croes GA. A method for solving traveling-salesman problems. Oper Res. 1958;6:791–812.

20. Englert M, Röglin H, Vöcking B. Worst case and probabilistic analysis of the 2-opt algorithm for the TSP. Algorithmica. 2014;68:190–264.

21. Xu X, Yuan H, Liptrott M, Trovati M. Two phase heuristic algorithm for the multiple-travelling salesman problem. Soft Comput. 2018;22:6567–81.

22. Kefi S, Rokbani N, Kromer P, Alimi AM. Ant supervised by PSO and 2-Opt algorithm, AS-PSO-2Opt, applied to traveling salesman problem. In: Proceedings of the IEEE International Conference; New York, NY, USA: Institute of Electrical and Electronics Engineers Inc; 2017. p. 4866–71.

23. Kent T, Richards A. Decentralised multi-demic evolutionary approach to the dynamic multi-agent travelling salesman problem. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Prague, Czech Republic, 2019.

24. Wooldridge MJ. The Logical Modelling of Computational Multi-Agent Systems. Ph.D. Thesis, University of Manchester, Manchester, UK; 1992.

25. Macal CM. Everything you need to know about agent-based modelling and simulation. J Simul. 2016;10:144–56.

26. Kazil J, Masad D, Crooks A. Utilizing Python for agent-based modeling: The Mesa framework. In: Social, Cultural, and Behavioral Modeling; Thomson R, Bisgin H, Dancy C, Hyder A, Hussain M, editors; Springer International Publishing: Cham, Switzerland; 2020. p. 308–17.

© 2026 The Author(s). | Published by Agro Intelligence Research Press