



Human–Machine Network Through Bio-Inspired Decentralized Swarm Intelligence and Heterogeneous Teaming in SAR Operations

Marc Espinós Longa¹ · Antonios Tsourdos¹ · Gokhan Inalhan¹

Received: 26 November 2021 / Accepted: 8 July 2022
© The Author(s) 2022

Abstract

Disaster management has always been a struggle due to unpredictable changing conditions and chaotic occurrences that require real-time adaption. Highly optimized missions and robust systems mitigate uncertainty effects and improve notoriously success rates. This paper brings a niching hybrid human–machine system that combines UAVs fast responsiveness with two robust, decentralized, and scalable bio-inspired techniques. Cloud-Sharing Network (CSN) and Pseudo-Central Network (PCN), based on Bacterial and Honeybee behaviors, are presented, and applied to Safe and Rescue (SAR) operations. A post-earthquake scenario is proposed, where a heterogeneous fleet of UAVs cooperates with human rescue teams to detect and locate victims distributed along the map. Monte Carlo simulations are carried out to test both approaches through state-of-the-art metrics. This paper introduces two hybrid and bio-inspired schemes to deal with critical scouting stages, poor communications environments and high uncertainly levels in disaster release operations. Role heterogeneity, path optimization and hive data-sharing structure give PCN an efficient performance as far as task allocation and communications are concerned. Cloud-sharing network gains strength when the allocated agents per victim and square meter is high, allowing fast data transmission. Potential applications of these algorithms are not only comprehended in SAR field, but also in surveillance, geophysical mapping, security and planetary exploration.

Keywords Swarm Intelligence · Decentralized Heterogeneous Systems · Bio-inspiration · Safe and Rescue · Monte Carlo Simulations · Human–Machine Network

1 Introduction

In the last decade, swarming has been a focus of research regarding SAR and disaster release operations, where time and resources are limiting factors. In real life, each scenario is unique and is conditioned by several operational aspects, e.g., communications, aerial regulations, maneuverability, power supply, situational awareness, processing power limitations, and so forth [1]. Semi-autonomous and fully autonomous swarming systems can enhance some of the prior working capabilities, while their design is heavily influenced by them.

Generally, the bottle neck of SAR activities falls upon the scouting stage, which commences after designating the boundaries of the observation area [2]. Harsh environmental restrictions often lead to improvisation and mission performance degradation [3]. Nonetheless, UAV searching mechanisms have proven to be very effective in reconnaissance tasks [4]. As long as targets are located gradually, rescuing phase is carried out in parallel. Due to the inherent high risk involved, this process requires intense or full human supervision and a series of complex sensors and actuators.

As far as recognition endeavors are concerned, task allocation and swarm functionality are crucial on performance. While centralized teaming offers finer global solutions, decentralized systems are more robust against communication loss and system failure, have greater adaptability levels under dynamic environments due to individual decision-making processes, and are highly scalable as a result of a decreasing computational demand [5]. Distributed probabilistic models framed under Markov Decision Process (MDP) or Partially Observable Markov Decision Process (POMDP)

✉ Marc Espinós Longa
marc.espinos-longa@cranfield.ac.uk;
a.tsourdos@cranfield.ac.uk; inalhan@cranfield.ac.uk

¹ School of Aerospace, Transport & Manufacturing (SATM), Cranfield University, Cranfield MK43 0AL, Bedfordshire, UK

have proven to be efficient in task allocation for multi-robot purposes with dynamic [6] and constrained tasks [7]. As reviewed in [8], nearly optimal solutions are achieved by probabilistic models [9] at a significant computational cost and time; hence the urge of precomputed searching routines [10–12]. Moreover, probabilistic models are very sensitive to the modelling phase [13]. For instance, in [14–16] Reinforcement Learning (RL) techniques are applied to control a group of agents in the classical hunter-prey pursuit game. Despite achieving optimal solutions for the given environment, swarming performance is strongly attached to hyperparameters and network configuration.

Decentralized bio-inspired systems have been applied successfully and widely in many areas such as problem optimization [17, 18] task allocation [19–22], Defense and Security sector [23], Artificial Neural Networks (ANN) optimization [24–26], path planning [27–29], searching activities [30] and fleets of UAVs [31, 32]. Based on stigmergy [33], i.e., indirect communication, biological swarming asynchronously places signs or cues in the environment to methodically complete tasks, e.g., ant pheromones for path optimization, bee structures for hive construction or bacterial substances for foraging activities. These attributes make those algorithms highly adaptable and robust. Nevertheless, some require prior environment information that usually is not available in SAR, e.g., Ant Colony Optimization (ACO) [24, 26] needs a predefined map with known paths. Despite relying on decentralized decision-making, other algorithms such as Particle Swarm Optimization (PSO) [34], based on flocking or schooling behaviors, use global information exchange for optimization, becoming poor candidates in wide exploration areas with low bandwidth schemes.

On the other hand, evolutionary algorithms inspired by Darwinian models of evolution (e.g., Coevolutionary Algorithm (COEA) [35–37] and Genetic Programming (GP) [38]) evaluate exhaustively candidate solutions through mutation mechanisms, i.e., arbitrary solution modifications. Although being efficient in small state spaces, evolutionary methods are intractable in large problems with time and computational constraints.

This piece of research introduces two techniques that regardless of their sub-optimal performance are highly responsive without prior environment information and utilize effective decentralized communications. Inspired by Bacterial and Honeybee Foraging behaviors [39, 40], CSN and PCN are solid, robust and scalable networks that rapidly adapt to big area extensions, dealing with critical scouting stages common in SAR missions.

The respective methodologies are adapted to a hybrid and heterogeneous team of UAVs and human rescue teams in a 2D post-earthquake scenario. Hybrid human-machine interaction grants rescue labors in parallel, mitigating the associated human risk. The main goal of the swarm is to obtain

victim coordinates and transmit the information to in-field rescuers. The study focuses on heterogeneous task allocation and team responsiveness under harsh communication schemes. Thus, target detection, tracking, path planning [41] (including collision avoidance), and communication protocols remain assumptions. The developed systems not only work for the studied case, but also can be applied to similar fields like planetary exploration, geophysical mapping, surveillance, security, and so forth.

The rest of the paper is organized as follows: in Sect. 2, global nomenclature and assumptions are reviewed. Section 3 introduces and describes CSN, including specific terminology, requirements, and discussion. Likewise, Sect. 4 reports PCN. Gathered results from Monte Carlo simulations are presented in Sect. 5, evaluating and comparing network performances for different parametrizations. Concluding marks and future work are presented in Sect. 6.

2 Problem Formulation

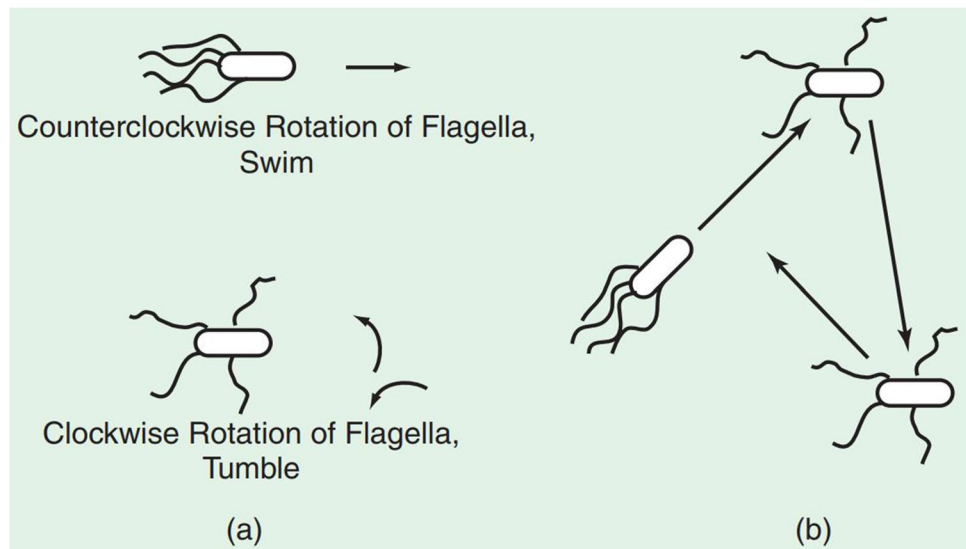
Let Λ be the vector representing the given space, being $\dim(\Lambda) = 2$, and A a set of heterogeneous agents, formed by t^A and α^A , with the respective control parameters σ_t^A and σ_α^A . t^A represents a group of UAVs with infrared (IR) sensors of sp_t detection range, capable of identifying victims located on the surface. On the other hand, α^A are UAVs equipped with acoustic sensors of sp_α detection range, used to reveal possible buried survivals. Positioning x, y , heading ϕ and flight mode ξ are common navigation attributes included in σ . Sensor measurements are given by C_t and C_α sensor fields, which are a linear representation of the scenario in bins b_t^A , being $\Lambda = \sum_{t=1}^n b_t^A$. Sensor fields contain all the information that agents can sense, including IR-visible victims, buried victims and rescuer signals.

All vehicles have duplexers, i.e., Transmit/Receive (TR) antennas, to send or receive specific coded signals either from A or P static¹ human rescue teams. Signal ranges are designated according to sp_A and sp_P specifications respectively. For a given k agent, strategic mission data is stored and transmitted through c_k information array. Victims are defined as $N = [v^N, \beta^N]$, being v^N IR-visible individuals located on the surface, and β^N buried people by the earthquake.

As far as mobility is concerned, $r_{k,i}(x_{k,i}, y_{k,i})$ represents movement constraints of k vehicle at time i , i.e., $r_{k,i}$ are no-flying zones that for the studied case delimit the designated area of operation. $m_{k,i}(r_{k,i}, \xi_{k,i}, \phi_{k,i})$ determines the movement of every vehicle. For a given optimal cruise speed u , flight

¹ Human teams are distributed in different interconnected static bases.

Fig. 1 Bacterial motion mechanism [39]



restrictions, flight mode and heading, $m_{k,i} = u(\cos\phi_{k,i}, \sin\phi_{k,i})$ decomposes movement in cartesian coordinates. Thus, considering unit time steps, next iteration position is computed $x_{k,i+1}, y_{k,i+1} = x_{k,i}, y_{k,i} + m_{k,i}(r_{k,i}, \xi_{k,i}, \phi_{k,i})$.

Initially, injured and rescue team bases are randomly allocated in Λ , while agents start from the cartesian origin located on the bottom-left corner. Despite having a two-dimensional space, this problem contemplates a third dimension where obstacle avoidance is assumed. The projection of agents' velocity in the x, y plane is assumed to be constant; hence, vehicles follow their 2D projection path without changing direction nor stopping. Simulation ends when all victim locations in Λ are discovered by the swarm and shared with human rescue teams. Ultimate goals are to minimize mission time and travel costs (refer to Sect. 5 Simulation Results for further detail). Remark that in a real-world case, human teams do not know the number of victims there are in a specific area. Thus, mission should not end until all surface and underground has been swept by the swarm. For this study, optimized area scanning has not been considered and remains as future work.

3 Cloud-Sharing Network

Bacterial Foraging Algorithm (BFA) [39] reproduces bacterial swarm exploratory behavior, moving towards food sources and avoiding noxious environments. Every time step, agents sense external stimulus and compute a gradient between previous and current measurements. Increments of food and toxicity levels act as attractive-repulsive forces, respectively. The overall decision-making process resembles to system control with integral feedback.

Bacterial motion mechanism alternates tumble and run, which are random and directional movements correspondingly. The first one is given for searching purposes, while the latter is used to reach strategic objectives. Thusly, an elevated percentage of runs occur when goals are nearby, whilst neutral or homogeneous surroundings entail higher exploratory moves. Bacterial motion, as described in Fig. 1, is achieved by means of the flagella; an organ placed at the rare end of the organism.

Stigmergic procedures are manifested as a form of cell-to-cell communication when external stressful agents are encountered. To overcome the adversity and enhance survivability, stimulating substances are generated to attract bacteria towards the epicenter of the invasion.

CSN incorporates gradient navigation and data-sharing through stimulating messages to gather valuable information in uncovered areas and spread it locally. Individual agents replicate and broadcast victim locations found by the swarm forming a cloud network that rapidly disseminates key rescue coordinates. Fast decentralized data transmission in dense agent systems makes CSN a powerful candidate in SAR operations. A general system diagram is displayed in Fig. 2, where the agent in red finds a victim location and broadcasts to other agents to reach human units. All vehicles, except those who found victims (guardians), are constantly moving to explore the area. Guardians are freed after victims are rescued by human.

3.1 Nomenclature and Requirements

For a set of agents r^A and α^A equipped with IR and acoustic sensors, let $\sigma_{I_k}^A$ and $\sigma_{\alpha_k}^A$ be an assortment of k vehicle control parameters including current position x_k, y_k , heading ϕ_k ,

Fig. 2 General Cloud-Sharing Network diagram

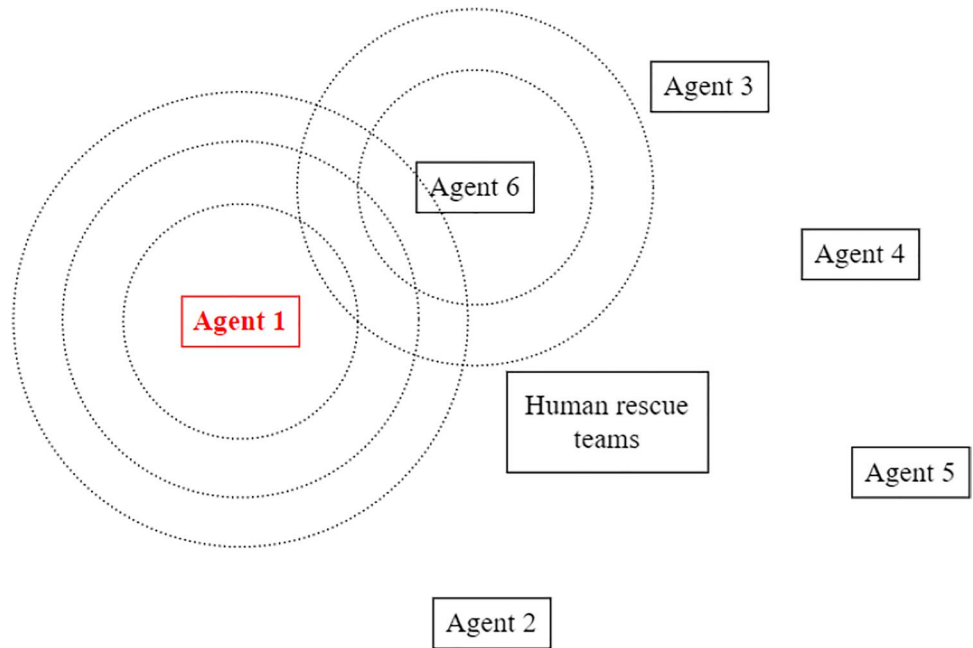
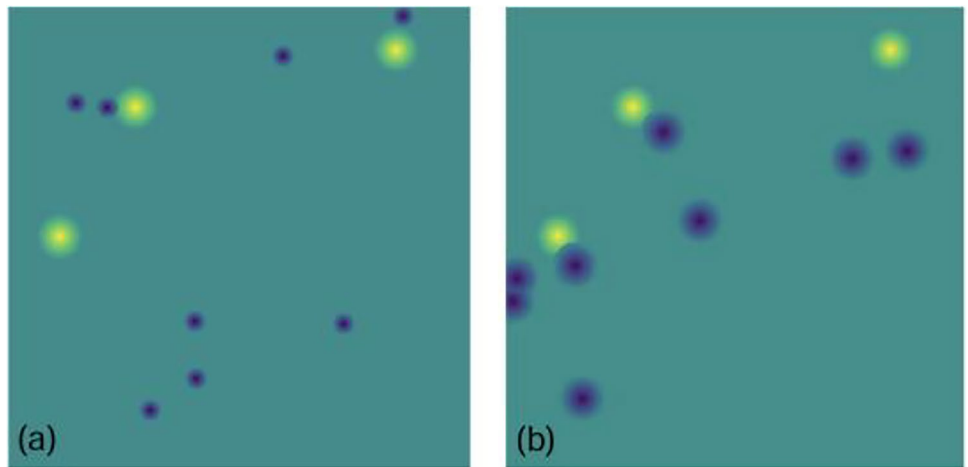


Fig. 3 Random agent sensor fields: (a) Acoustic-TR sensor field, (b) IR-TR sensor field



flight mode ξ_k , tumble- run alternation rate τ_k , and previous position $x_{k,prev}, y_{k,prev}$.

CSN incorporates a navigation gradient-based method similar to potential field. Victims play an attractive role adopting negative values in C_i and C_α sensor fields, while signals emitted by human rescue teams and eventual agents are seen as deflectors and are interpreted with positives. The latter are remarkably important, agent deflecting signals are used for task relocation, avoiding more than one vehicle getting stuck in a same location. Likewise, P signals avert agent surveillance near human teams for rescue efficiency matters.

Figure 3 yellow areas of influence, identical in both samples, are signals received from rescuer teams through TR antennas. Purple areas represent sensor measurement from acoustic and IR sensors. Note that acoustic measurements have smaller radius due to specifications. At i time step,

$\Delta g_{k,i}(C_{k,i}, C_{k,i-1})^2$ sensor gradient is computed for every k agent, then flight mode $\xi_{k,i}(\Delta g_{k,i})$ is selected accordingly. If measurements surpass threshold $\gamma < 0$, a victim is close enough to be acquired by the respective agent.

To accelerate convergence and avoid getting stuck in different flight modes when computing $\Delta g_{k,i}$, state aggregation techniques are applied, dividing $\phi \in 2\pi$ into 8 intercardinal circular sectors $\theta = (\theta_N, \theta_{NE}, \theta_E, \theta_{SE}, \theta_S, \theta_{SW}, \theta_W, \theta_{NW})$ of $\pi/4$ radians. Whenever there is a $r_{k,i}$ mobility restriction, the whole section $\theta_{k,i}$ containing $\phi_{k,i}$ is banned. However, in normal conditions, vehicles can take any $\phi_{k,i} \in 2\pi$.

² C_k is equivalent to C_i or C_α depending on k vehicle type. For simplicity, this notation is used multiple times with different algorithm parameters (e.g., sp_k instead of sp_i or sp_α , σ_k^A instead of σ_i^A or σ_α^A).

Tumble-run alternation rate $\tau_{k,i}$ is used as an exploring tool to compute $\phi_{k,i}$. $\tau_{k,i} = 1/100$ means that for every 100 runs where $\phi_{k,i} = \phi_{k,i-1}$ if $\phi_{k,i-1} \notin r_{k,i}$, the agent performs a tumble, in other words, randomly selects a non-restricted heading $\phi_{k,i} \notin r_{k,i}$. If runs are interrupted by flight restrictions $r_{k,i}$, agent tumbles and the process restarts.

Communication array of information $c_{k,i}$ contains all the victim locations found by agent k or received from other agents. Every time an agent receives a sequence of information from a close co-worker with new victim locations, the local data array is updated, and broadcasting labors resume. Hence, agents constitute a cloud-sharing network, where data is spread locally through the swarm until reaching P teams. Survivals located by P are contained in Ψ_i .

Algorithm After initializing all main simulation objects, first C_k sensor fields are obtained and main simulation loop in Fig. 4, is triggered. Every dt , each k agent measures and computes a sensor gradient.

$$\Delta g_{k,i}(C_{k,i}, C_{k,i-1}) = C_{k,i} - C_{k,i-1}, \tag{1}$$

being $C_{k,i} = C_k(x_{k,i}, y_{k,i})$ the current sensor measurement, and $C_{k,i-1} = C_k(x_{k,i-1}, y_{k,i-1})$ the previous sensor measurement. Depending on $C_{k,i}$ and $\Delta g_{k,i}(C_{k,i}, C_{k,i-1})$, k agent switches to a flight mode $\xi_{k,i}(C_{k,i}, \Delta g_{k,i})$.

Guarding victim ($C_{k,i} \leq \gamma$) If current measurement in bin $b_{k,i}^A$ is smaller than γ target detection threshold, k agent automatically activates this mode, acquiring the location of the victim and entering in hovering state. Subsequently, a deflecting signal is generated affecting $C_{t,i}$, $C_{\alpha,i}$ sensor fields. In addition, a stimulating message with the target position is broadcasted in $c_{k,i}$ communication array. Therefore, this temporary mode initiates two noteworthy mechanisms: stigmergic task relocation through temporary deflecting signals (i.e., pulses generated by vehicles act as an eventual repulsive cue placed in the environment, affecting the behavior of the swarm), and cloud-sharing communication where agents generate local messages with victim positions.

Like repeaters, in-range nearby vehicles intelligently select, replicate, and add new data sequences to $c_{k,i}$ array. Together, both stratagems ensure an efficient and decentralized data transmission and task allocation. Guarding victim flight mode is active until the covered location is acquired by any rescue team and subsequently included in Ψ_i . At that moment, the victim is considered rescued, the deflecting signal of k vehicle ceases, and exploring mode is reactivated. Each time step, P rescue teams check the communication cloud to look for new ν^N, β^N target positions. Simulation ends when $\nu^N, \beta^N \in \Psi_1$ (when all victims are localized by P).

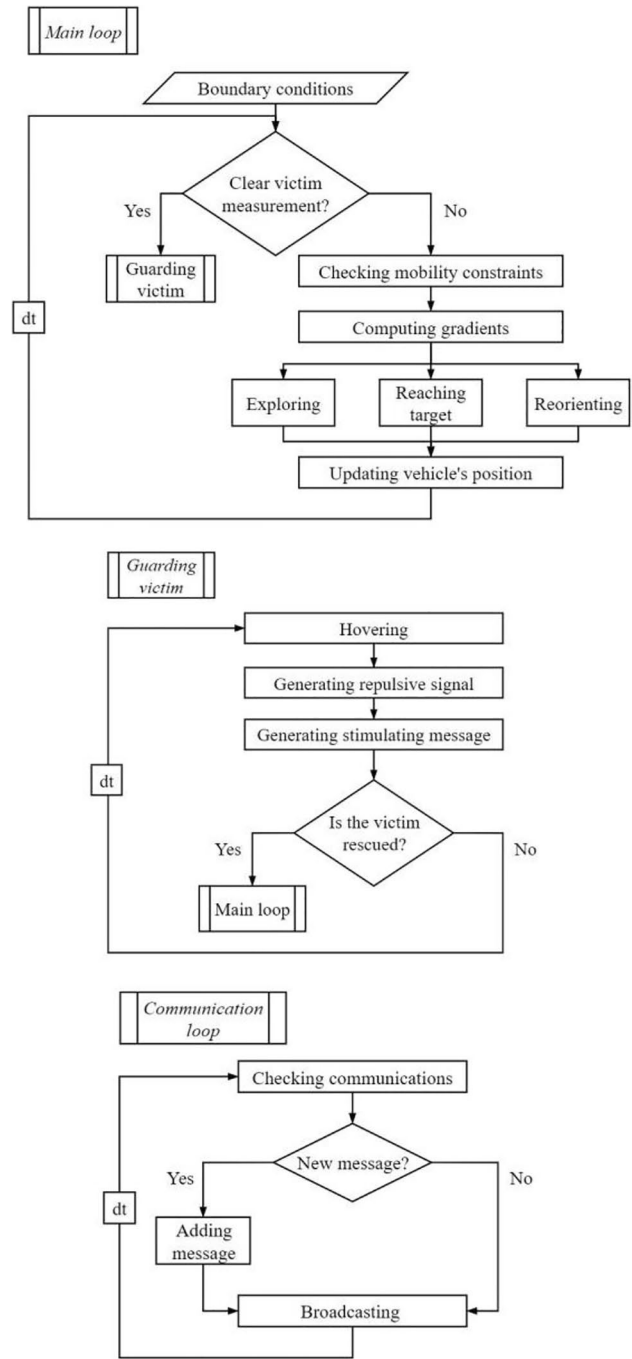


Fig. 4 CSN main simulation loops: (a) Main loop, (b) Guarding victim, (c) Communication loop

Exploring ($\Delta g_{k,i} = 0$) Agent k keeps searching at $\tau_{k,i}$ tumble and run alternation rate. This stochastic process does not signify a big computational challenge.

Reaching target ($\Delta g_{k,i} < 0$) At time i , heading $\phi_{k,i} = \phi_{k,i-1}$, i.e., if the gradient measured is negative, the agent keeps with the previous heading. Hence, agents are progressively oriented towards acquisition targets.

Table 1 CSN simulation

Initialize landscape Λ , agents $l, \alpha \in A$, victims $v, \beta \in N$, and rescue teams P objects (sensor detection and signal ranges sp included)
Distribute randomly N, P in Λ
Initialize C_l, C_α sensor fields and rescued victims Ψ
Initialize agent positions, comms arrays c , and tumble-run alternation rate τ
Loop for each time step ($i = 0, 1, \dots, T - 1$):
Loop for each agent ($k = 0, 1, \dots, A - 1$):
Add new in-range messages to local comms $c_{k,i} \leftarrow c_{k,i} + c_i(x_{k,i}, y_{k,i} - \sigma_i^A(x_i, y_i) \leq sp_A)$
Get flight constraints $r_{k,i} \leftarrow x_{k,i}, y_{k,i}$
Compute measurement gradient $\Delta g_{k,i} = C_{k,i} - C_{k,i-1}$
If $C_{k,i} \leq \gamma$ or $\xi_{k,i} = \text{Guarding victim}$:
$\xi_{k,i} \leftarrow \text{Guarding victim}$
Add located victim to local comms $c_{k,i} = c_{k,i} + N_k$
Generate deflecting signal $C_{k,i} \leftarrow x_{k,i}, y_{k,i}, sp_k$
If N_k in Ψ_i :
$\xi_k = \text{Exploring}$
Elif $\Delta g_{k,i} = 0$:
$\xi_{k,i} \leftarrow \text{Exploring, compute heading } \phi_{k,i} \text{ from } \tau_k$
Elif $\Delta g_{k,i} < 0$:
$\xi_{k,i} \leftarrow \text{Reaching target, maintain heading } \phi_{k,i} = \phi_{k,i-1}$
Elif $\Delta g_{k,i} > 0$:
$\xi_{k,i} \leftarrow \text{Reorienting}$
Change direction randomly fulfilling $\phi_{k,i} \neq \theta_{k,i-1}$
Store previous position $x_{k,i-1}, y_{k,i-1} \leftarrow x_{k,i}, y_{k,i}$
Hovering displacement $m_{k,i} = 0$ if $\xi_{k,i} = \text{Guarding victim}$, else $m_{k,i} = u(\cos \phi_{k,i}, \sin \phi_{k,i})$
Update position $x_{k,i}, y_{k,i} \leftarrow x_{k,i}, y_{k,i} + m_{k,i}$
Update parameters $\sigma_{k,i}^A \leftarrow x_{k,i}, y_{k,i}, \phi_{k,i}, \xi_{k,i}, x_{k,i-1}, y_{k,i-1}$
Loop for each rescue team ($p = 0, 1, \dots, P$):
Add new in-range victim location messages $\Psi_i \leftarrow \Psi_i + c_i(x_p, y_p - \sigma_i^A(x_i, y_i) \leq sp_A)$
If $\Psi_i = N$:
Terminal state $i = T$

Reorienting ($\Delta g_{k,i} > 0$) Either there is a deflector point nearby (rescuer or agent “noxious” signal) or agent k is flying away from a victim. In any case, heading is switched randomly, fulfilling $\phi_{k,i} \neq \theta_{k,i-1}$ for some $\phi_{k,i-1} \in \theta_{k,i-1}$; that is, any direction not included in the previous circular sector. Thence, positive gradients (1) unleash position adjusting techniques towards strategical points.

Formal definition of CSN is presented in Table 1. Note that T time step is reached on a terminal state when simulation ends. New in-range messages imply unduplicated positioning data, i.e., vehicles and rescue teams smartly select new data sequences in the cloud, absorb recent information and recodify their own transmission. Positions $\sigma_i^A(x_i, y_i)$ comprehend all vehicle locations at time step i , while sp_A represents the beacon range of A agents (refer to Sect. 2 Problem Formulation for further notation detail). After finding a victim N_k and incorporating its position within $c_{k,i}$, a deflecting signal of range sp_k is generated, triggering an update of the global sensor field $C_{k,i}$. In terms of convergence, highlight the state aggregation technique used in reorienting flight mode to escape gradient oscillations (from $\Delta g_{k,i} > 0$ to $\Delta g_{k,i} < 0$

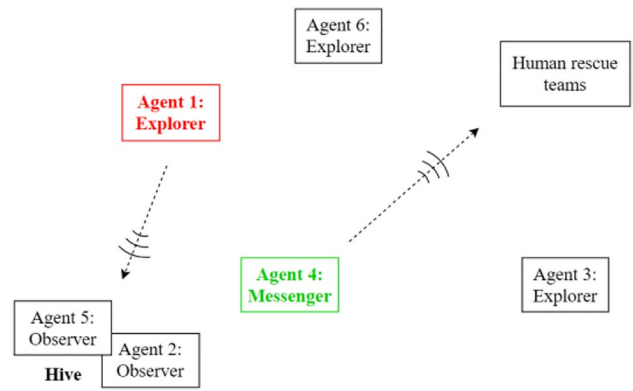


Fig. 5 General Pseudo-Central Network diagram

and vice versa). $\theta_{k,i-1}$ heading restrictions provide vehicles enough inertia to break non-transient states.

4 Pseudo-Central Network

Bees are known to live in hierarchical structures with heterogeneous roles, all contributing to the survival of the hive. Many algorithms have been inspired by these little co-workers. This paper presents PCN, specifically inspired in *Honeybee Foraging Algorithm* (HBFA) [40].

Nectar recollection starts when explorer bees leave the hive to search profitable sources while observer bees stay passively. After finding a site, explorer co-workers return to the hive with a little sample of nectar, and a surprisingly advanced communication process starts. Explorer bees carry out an 8-shaped dancing performance at a specific angle from the vertical of the colony. From the wagging performance, observer bees deduce the direction of the strategic location by measuring the angle of the dancing pattern with respect to the sun. It has been demonstrated that foraging bees modify the angle of their appearance throughout the day.

Moreover, passive bees estimate the amount of nectar found, as it is directly linked to the time and strength in which their swarm mates dance. Bigger foraging key points are associated with impetuous and lasting performances, which consequently recruit greater number of individuals to exploit all natural resources. Profitability of a site is reduced as the number of bees exploiting its nectar increases. Nectar quality can always be tested with samples that explorer bees carry themselves.

PCN takes advantage of honeybee organization structure, using the hive as a pseudo-central node to locally communicate with the rest of the swarm. Each agent, after finding any point of interest (victims or rescue teams), returns to the hive to share the information with the structure as the explorer agent in red indicates in Fig. 5. All passive vehicles (observers) that are connected to that

node have access to the recollected data, and head towards human rescue teams (messenger) to share information. The proposed honeybee network becomes strong in weak communication scenarios with vast ground extensions, a perfect fit for the studied case.

4.1 Nomenclature and Requirements

Let σ_k^A represent the main parameters of k agent with current position x_k, y_k , heading ϕ_k , flight mode ξ_k and observer parameter o_k counting the amount of time steps that the vehicle passively remains at H hive node located on the cartesian origin. Agents resting at H make PCN compatible with autonomous dock and advanced recharging systems [42, 43], an option worth considering enhancing swarm autonomy. At i time step, the hive contains $h_{p,i}, h_{N,i}$ rescuer and victim locations shared by previous explorers, i.e., initially agents are not aware neither of injured nor human team positions. Therefore, explorers need to identify these coordinates and share them with the colony. To distinguish victims on the surface from human rescuers, P emit an encoded signal of sp_p range with their position.

Another important parameter of PCN is probability distribution $p_{k,i}(h_{N,i}, v^N, \beta^N, \Psi_i)$ (2). For a given k vehicle at i time step, $p_{k,i}$ is the chance to head towards a rescuer team, if acquired, and share new victim locations contained in H .

$$p_{k,i} = \frac{a_i}{\rho_i} = \frac{h_{N,i} - \psi_i}{\bar{\psi}_i} \text{ if } h_{p,i} > 0, \quad (2)$$

else $p_{k,i} = 0$

$\bar{\psi}_i = v^N + \beta^N - \psi_i$ are the remaining unlocated victims, and $a_i(h_{N,i}, \psi_i)$ the relative new positions stored by the hive. Notice the difference between Ψ_i and ψ_i ; while the first term refers to injured coordinates obtained by P, the latter are victim locations sent through an agent to a rescuer team. Consequently, ψ_i can immediately be updated after recruiting an agent to head P, and $a_i = 0$. Additionally, $p_{k,i}$ is 0 if no human team positions have been gathered by H , that is, $h_{p,i} = 0$. Greater number $a_i(h_{N,i}, \Psi_i)$ increases the probability of being recruited, as a passive vehicle, to share new data with P.

For efficient agent and energy management, note that initially $\bar{\psi}_i \gg a_i$. Thereupon, recruitment usually occurs when H has gathered a significant number of coordinates. Besides, recruited agents always fly to the closest available $h_{p,i}$ location. Any vehicle staying at H more than θ_o time limit switches to explorer mode and o_k resets. Hence, active and passive swarm balance is dynamically modified according to environment needs.

Unlike CSN, PCN place physical cues (e.g., paint, radio beacons) on top of target locations to warn other co-workers that respective tasks have been completed and enhance swarm resource allocation. In a real-world case, this process may be arduous and time consuming compared to CSN task relocation

mechanism. Different indirect communication techniques are suggested to be tested in real experiments. Likewise, navigation is not gradient-based, and point acquisitions happen instantly after discovery. In compensation, the given analysis considers smaller sp_i and sp_α victim detection ranges in PCN.

4.2 Algorithm

Initially, primary agent roles are distributed heuristically and PCN exploring and observing swarm behaviors in Fig. 6 ensue. Agent $\xi_{k,i}(C_{k,i})$ flight modes are selected based on sensor measurements.

Exploring ($C_{k,i}(x_{k,i}, y_{k,i}) = 0$) The agent selects randomly heading ϕ_k and follows it until flight restrictions require a direction adjustment. The implemented searching stage, presented as an alternative approach, has fewer turn rates (directional adjustments) per agent than CSN.

Returning Hive ($C_{k,i}(x_{k,i}, y_{k,i}) < 0$) Agent k finds a point of interest (i.e., victim or rescue team), stores it in $c_{k,i}$ communication array, places a physical cue in Λ to alert the swarm that the respective site has already been attended ($C_{k,i}$ sensor field is modified), and computes the vector back to H node each iteration. When H is reached and, the agent shares the information with the hive and switches to observing state.

Observing (Reached H) Agent awaits new locations to be shared with the hive by foraging vehicles. Afterwards, the recruitment stochastic process takes place following $p_{k,i}$ probability distribution defined in (2). If the agent is recruited flight mode changes to heading rescuer. On the other hand, if $o_{k,i} \geq \theta_o$, observer shifts to exploring mode.

Heading Rescuer (Agent Recruited) Flying towards the closest $h_{p,i}$ rescue team location. Optimal path is designed to achieve an energy-efficient swarm behavior. After reaching the target, agent shares the information with P, increasing Ψ_i rescued victims, and exploring stage starts again.

Table 2 displays a formal definition of PCN algorithm. In a real experiment, v^N and β^N existing victims are unknown. Approximations based on current disaster scenario are suggested to be used instead.

5 Experimental Results and Discussion

CSN and PCN systems are evaluated and compared under a SAR framework. The proposed metrics, state-of-the-art parameters in disaster release operations, assess algorithm and mission performance.

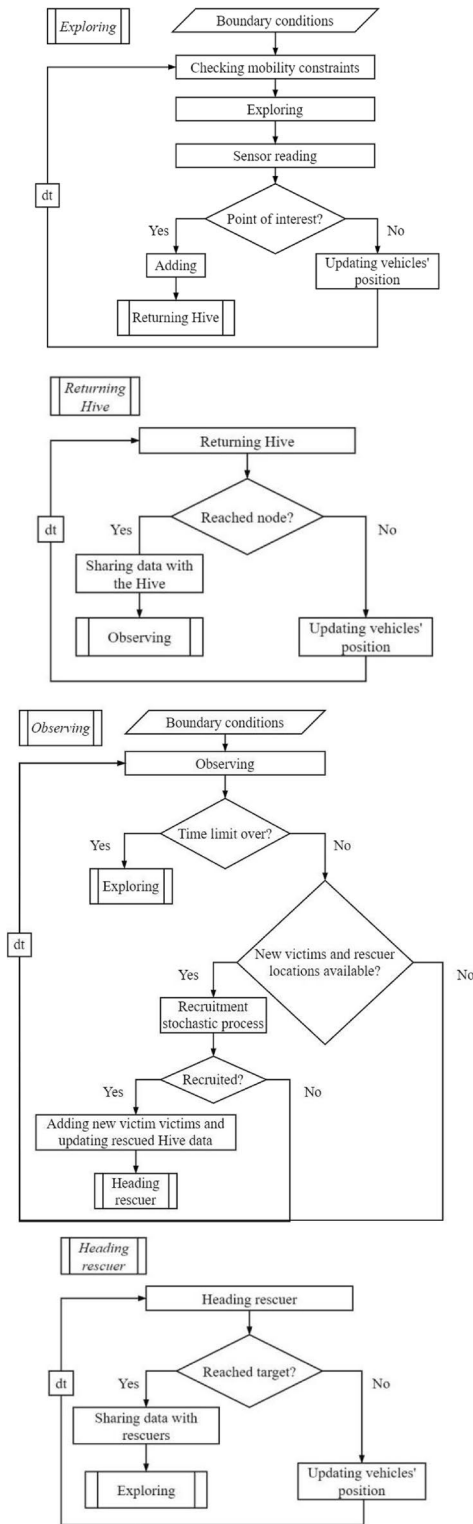


Fig. 6 PCN main simulation loops: (a) Exploring, (b) Returning Hive, (c) Observing, (d) Heading rescuer

Time-to-complete mission (TTC) [44, 44] focuses on rescue efficiency. Expressed in seconds, represents the time in which human rescue teams acquire all victim coordinates.

As far as autonomy and energy saving are concerned, travel efficiency (TE) and travel per victim captured per vehicle (TVV) are employed. As a modification of path length [46, 47] TE is the minimum path divided by the total travelled distance, where the first term is twice the distance from all points of interest (victims and human team bases) to the cartesian origin, and the latter refers to the total meters flown by agents. Instead, TVV are the meters travelled by each agent divided by the number of victims allocated in the scenario.

5.1 Fleet Size Analysis

Algorithm performance is tested in Fig. 7 for different swarm sizes, where scenario dimension $\Lambda = 1 \text{ km}^2$, victims $N = 16$, and human rescue bases $P = 3$.

Both algorithms present similar tendencies for an increasing number of agents, with logarithmic decrements in all the analysis. PCN demonstrates to be an efficient method for a reduced number of agents, ergo, in environments with lower number of allocated vehicles per area unit. Despite facing a hard scouting phase, hive communication system and path optimization techniques allow agents to complete their task one third quicker than CSN for a swarm of 10 agents.³ However, exploring stochasticity leads travel efficiency to dramatically decrease when the number of agents per area rises. PCN path optimization weakens and TVV aligns with CSN energy consumption.

Timewise, both algorithms enhance performance as the swarm scales up in number. Cloud-sharing scheme empowers in dense swarms. Thus, fast-data transmission authorizes CSN to outperform the Hive for big fleets of 70 agents.

5.2 Scenario Dimension Analysis

The studied networks are examined in different coverage areas, given a fixed number of vehicles $A = 50$, victims $N = 50$, and rescuers $P = 3$.

As expected, vaster coverage areas in Fig. 8 result in slower rescue performance and greater energy consumption for a fixed swarm size. Low in-field agent density makes CSN struggle to perform in big areas, while PCN pseudo-central communication method mitigates considerably ground extension effects. Thence, the difference between CSN and PCN mission time growth. Travelled meters per victim and agent rise similarly, inducing a decrease of displacement efficiency.

³ Agents are distributed equally in r^A and α^A heterogeneous vehicles (e.g., 40 agents are composed by 20 r^A and 20 α^A vehicles).

Table 2 PCN simulation

```

Initialize landscape  $\Lambda$ , agents  $\iota, \alpha \in A$ , victims  $v, \beta \in N$ , and
rescue teams P objects (sensor detection and signal ranges  $sp$ 
included)
Distribute randomly N, P in  $\Lambda$ 
Initialize  $C_\iota, C_\alpha$  sensor fields and rescued victims  $\Psi$ 
Initialize agent positions, role distribution (exploring and
observing flight modes  $\xi$ ), headings  $\phi$ , observing counter  $o$ ,
time limit  $\theta_0$  comms array  $c$ , and Hive data  $H(h_{P,i}, h_{N,i}), \psi$ 
Loop for each time step ( $i = 0, 1, \dots, T - 1$ ):
  Loop for each agent ( $k = 0, 1, \dots, A - 1$ ):
    Get flight constraints  $r_{k,i} \leftarrow x_{k,i}, y_{k,i}$ 
    If  $\xi_{k,i} = \text{Exploring}$ :
      Check measurements  $C_{k,i}$ 
      If  $C_{k,i} = 0$ :
        Maintain heading  $\phi_{k,i} = \phi_{k,i-1}$  if  $\phi_{k,i-1} \notin r_{k,i}$ , else
        select random  $\phi_{k,i}$ 
      Else:
        Add point of interest to  $c_{k,i}$  (victim or rescuer)
        Switch to  $\xi_{k,i} = \text{Returning Hive}$ 
        Place cue at current position (recompute  $C_{k,i}$ )
        Compute heading  $\phi_{k,i}$  to  $H$ 
    Elif  $\xi_{k,i} = \text{Returning Hive}$ :
      Check distance to  $H$ 
      If  $|H - x_{k,i}, y_{k,i}|$  is small:
        Switch to  $\xi_{k,i} = \text{Observing}$ 
        Add  $c_{k,i}$  data to  $H(h_{P,i}, h_{N,i})$ 
      Else:
        Compute heading  $\phi_{k,i}$  to  $H$ 
    Elif  $\xi_{k,i} = \text{Observing}$ :
      If  $o_{k,i} \geq \theta_0$ :
        Switch  $\xi_{k,i} = \text{Exploring}$ , reset  $o_{k,i}$ 
      Else:
        Compute  $p_{k,i} = (h_{N,i} - \psi_i) / (v^N + \beta^N - \psi_i)$  if
        available rescue teams  $h_{P,i} > 0$ , else  $p_{k,i} = 0$ 
        Evaluate recruitment stochastic process
        If agent recruited:
          Switch  $\xi_{k,i} = \text{Heading rescuer}$ 
          Compute heading  $\phi_{k,i}$  to nearest rescuer
          Store new victim points in  $c_{k,i} = h_{N,i} - \psi_i$ 
          Update  $\psi_i$ , reset  $o_{k,i}$ 
        Else:
           $o_{k,i} = o_{k,i} + 1$ 
    Elif  $\xi_{k,i} = \text{Heading rescuer}$ :
      Check distance to P
      If  $|P - x_{k,i}, y_{k,i}| < sp_A$ :
        Switch to  $\xi_{k,i} = \text{Exploring}$ 
        Update rescued victims  $\Psi_i$ 
      Else:
        Computed heading  $\phi_{k,i}$  to P
    No displacement  $m_{k,i} = 0$  if  $\xi_{k,i} = \text{Observing}$ ,
    else  $m_{k,i} = u(\cos\phi_{k,i}, \sin\phi_{k,i})$ 
    Update position  $x_{k,i}, y_{k,i} \leftarrow x_{k,i}, y_{k,i} + m_{k,i}$ 
    Update parameters  $\sigma_{k,i}^A \leftarrow x_{k,i}, y_{k,i}, \phi_{k,i}, \xi_{k,i}, o_{k,i}$ 
  If  $\Psi_i = N$ :
    Terminal state  $i = T$ 
    
```

Disparity in results is another noteworthy aspect to consider. PCN confidence intervals around medians are especially smaller than in CSN, ergo, the Hive achieves a constant and stable performance independently of the initial randomized problem distribution. Hence, cloud-sharing system becomes vulnerable and environment-dependent in low-density networks.

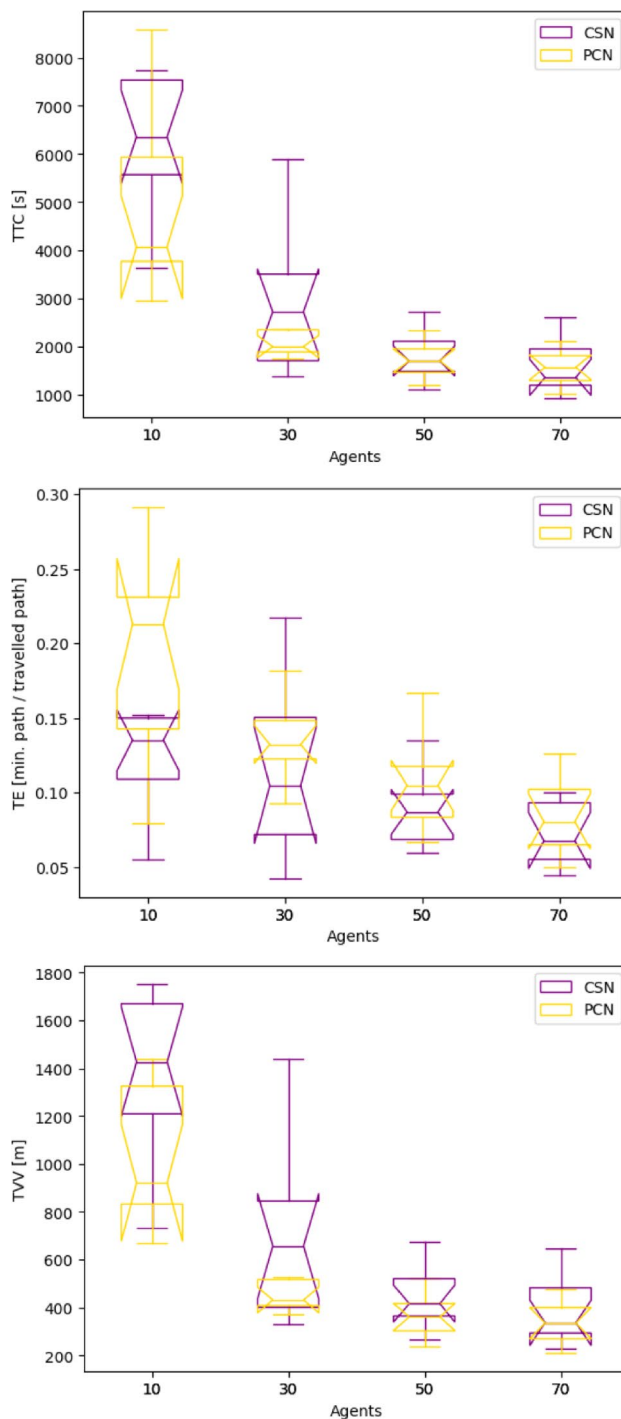


Fig. 7 Fleet size simulation analysis

5.3 Victim Number Analysis

In SAR operations, victim density fluctuates according to the occurrences. Allocated rescue resources and deployed systems need to be dynamic to fulfill mission requirements. Thus, it is important to study CSN and PCN performance

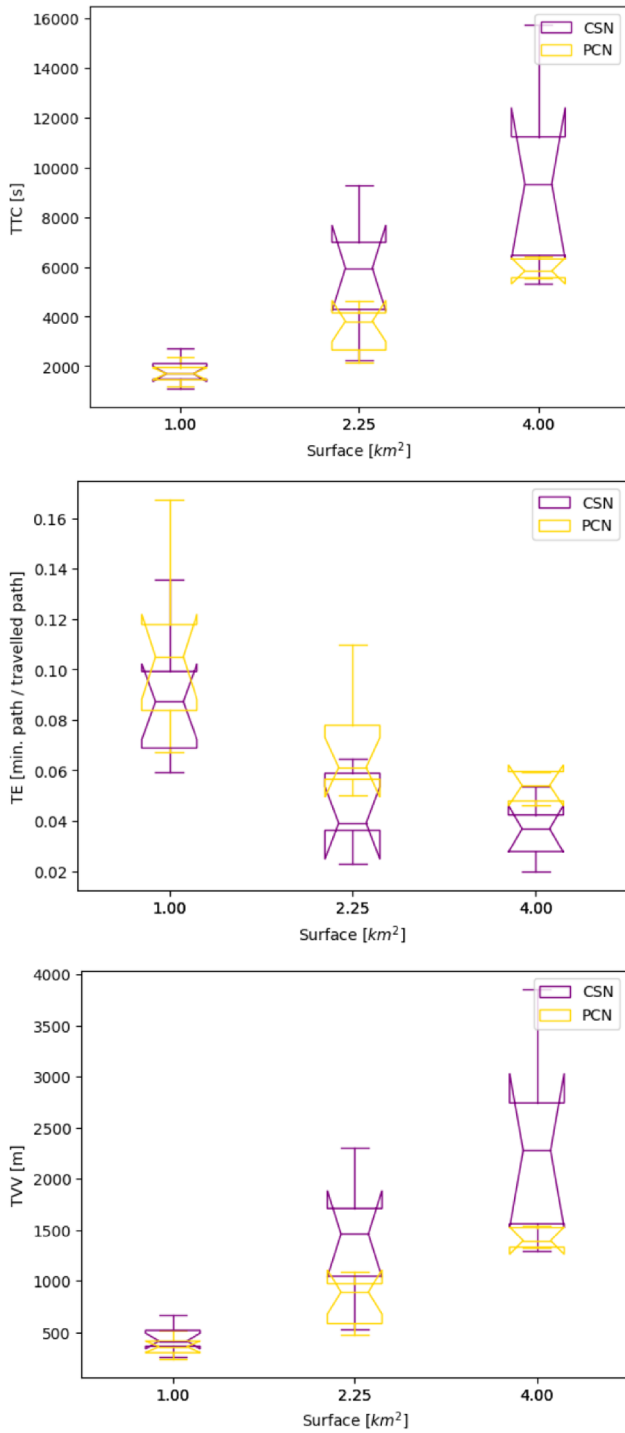


Fig. 8 Scenario dimension simulation analysis

tendencies for a variable number of injured.⁴ The rest of environment and algorithm parameters are $\Lambda = 1 \text{ km}^2$, $A = 50$, and $P = 3$.

⁴ Like agents, victims are distributed evenly in surface v^N and buried β^N targets.

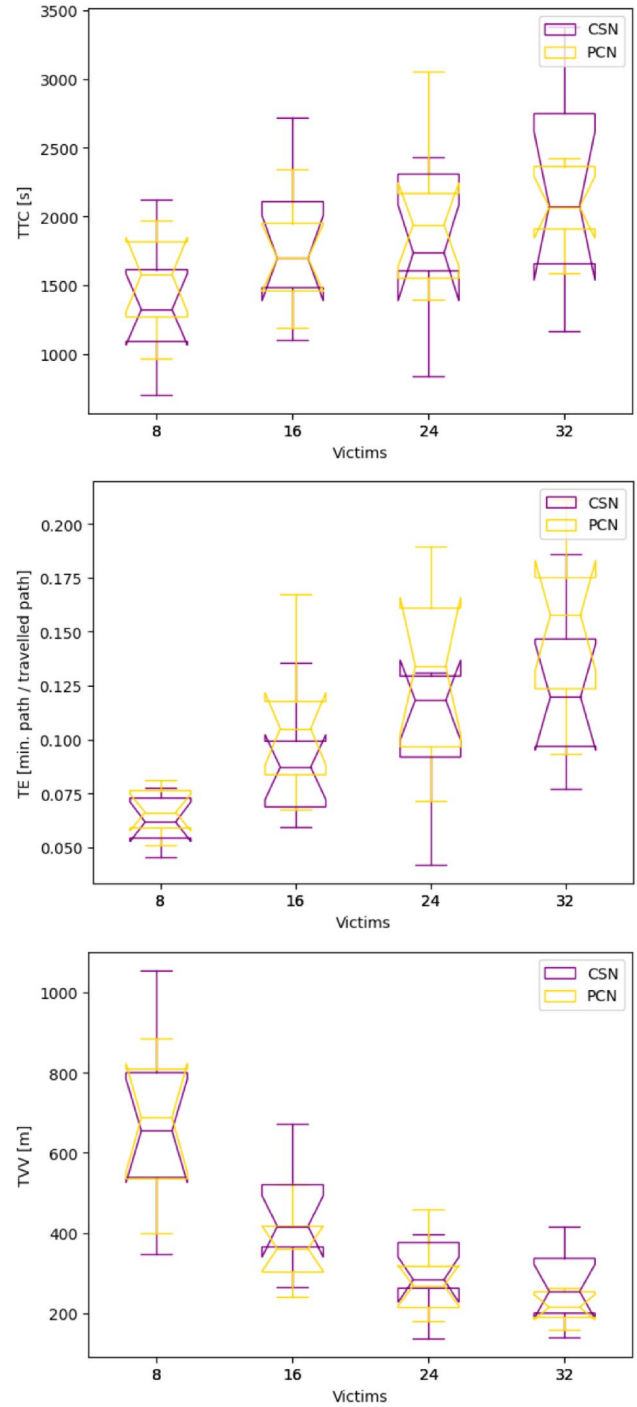


Fig. 9 Victim number simulation analysis

Dealing with an elevated number of tasks is often time and energy consuming. However, Fig. 9 displays an energetic improvement in both human-machine networks. Despite showing analogous performances, observing agents in PCN accomplish greater energy savings by waiting in H a considered amount of time before sharing information with closest P human camps. Therefore, the colony obtains lower

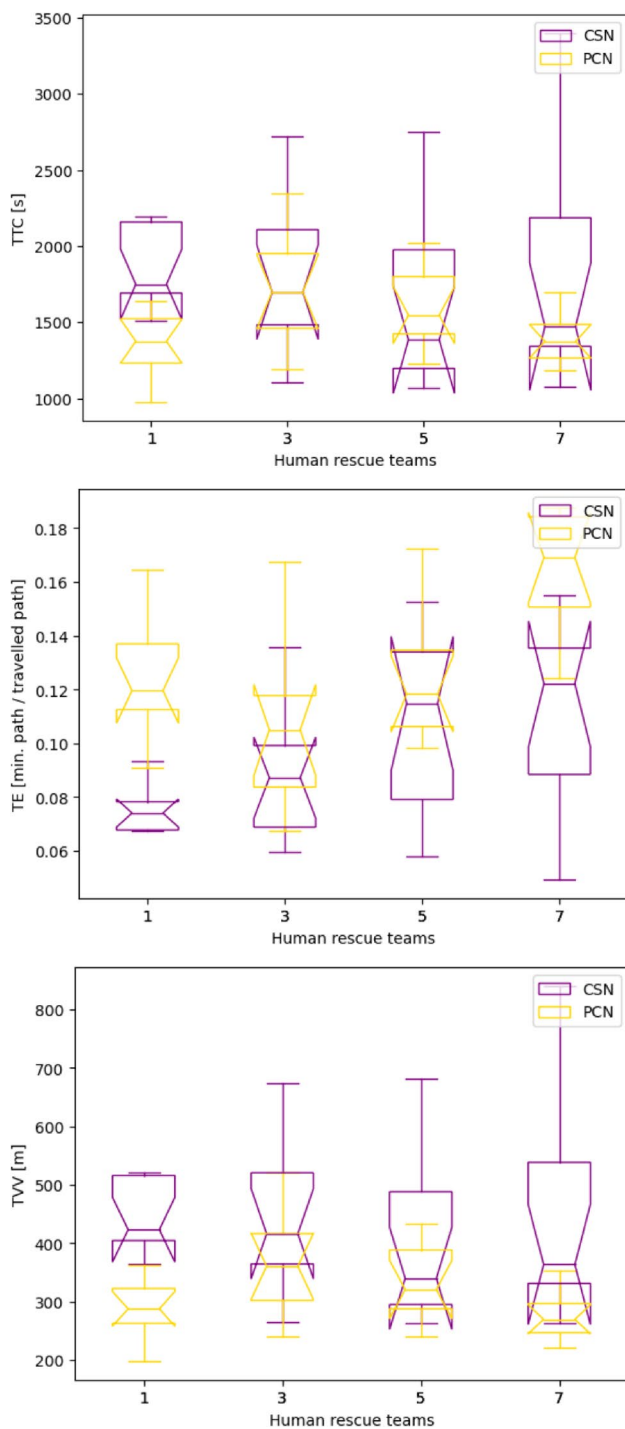


Fig. 10 Variable human teams simulation analysis

TVV and greater TE for an increasing number of injured. Furthermore, role balance effectively manages vehicle allocation in scouting stage, attaining similar results to CSN in terms of simulation time.

On the other hand, the cloud performs accurately for a low number of victims. In this manner, CSN is not only affected by agent-extension density, but also by victim-agent relation.

A reduced number of victims with respect to agents allow to allocate more vehicles as scouters, and as a result, victims are located faster, and data is spread quicker towards P.

5.4 Variable Human Teams Analysis

P rescue teams are key data-sharing points that play an essential role in both the SAR operation and human–machine connection. Figure 10 reflects the influence of these strategic coordinates, in terms of quantity, for a set of $A = 50$, $N = 16$, and a coverage area of $\Lambda = 1 \text{ km}^2$.

As expected, cloud rescue and energetic performance is significantly enhanced when augmenting P. This way, agents need not travel that much to share victim positions. Given a set of $P = 5$, CSN towers the Hive regarding mission time, and almost reaches PCN in energy optimization endeavors. However, an excessive number of human bases appears to have a negative impact on CSN performance. What at first seemed like a capable task relocation mechanism, human deflecting signals end up being an obstacle to the swarm, especially affecting exploring vehicles.

Contrarily, an opposite effect develops in the colony. From previous analysis, PCN already demonstrates outstanding outcomes in low-density areas, that is, in vast coverage areas and small fleets. Likewise, the colony remarkably outreaches the cloud for $P = 1$. Nonetheless, performance, in all aspects, is degraded in the subsequent batch. PCN agents need first to locate human team positions before sharing target coordinates. This fact becomes a short-term drawback that manifests its peak when $P = 3$. Onwards, path-optimization mechanism improves with human teams, and so does PCN behavior.

6 Conclusion and Future Work

The research conducted in this paper aims to contribute and enhance SAR operations by coupling swarming and human rescue systems. The gathered results attempt to be a step forward in real human–machine hybrid solutions for disaster release operations, becoming a niching reference in future research.

CSN and PCN demonstrate to be dynamic, decentralized, scalable, and semi-autonomous schemes promoted in bigger swarm sizes. Cloud-sharing network develops a powerful performance when the allocated agents per victim and squared meter is high. Pseudo-central network exhibits a solid and consistent performance throughout the entire study. Hive decentralized communication combined with path-optimization techniques unfold an exceptional execution in large areas and diminished systems. Moreover, PCN proves to be energetically superior to CSN in all the analysis, i.e., travel efficiency is higher and flight distance is curtailed.

Both CSN and PCN fulfill operational demands despite achieving sub-optimal performance.

Future work considers optimized searching techniques to improve scouting and artificial intelligence, such as Deep Reinforcement Learning (DRL) methods, to enhance swarming. In addition, a realistic analysis in 3D physics simulators is likely to be considered.

Authors Contributions All authors contributed to the study conception and design. Literature review, mathematical development and analysis were performed by Marc Espinós Longa. The first draft of the manuscript was written by Marc Espinós Longa and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) and BAE Systems under the project reference number 2454254.

Data Availability The presented manuscript has no associated data and does not contemplate future depositions.

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

Ethics Approval This is a software simulation study. No human or animal subjects are related to this study. Hence, ethical approval is not applicable.

Consent to Participate This is a software simulation study. No human subjects are involved in this study. Therefore, consent to participate is not applicable.

Consent to Publish This is a software simulation study. No human subjects are involved in this study. Therefore, consent to publish is not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Chen, D., Liu, Z., Wang, L., Dou, M.: Natural disaster monitoring with wireless sensor networks: a case study of data-intensive applications upon low-cost scalable systems. *Mobile Networks and Applications* **18**(5), 651–663 (2013). <https://doi.org/10.1007/s11036-013-0456-9>
- Office, U. N.: INSARAG Guidelines and Methodology, p. 150 (2007).
- Ochoa, S. F. and Santos, R. (2015) “Human-centric wireless sensor networks to improve information availability during urban search and rescue activities,” *Information Fusion*. Elsevier B.V., 22, pp. 71–84. <https://doi.org/10.1016/j.inffus.2013.05.009>.
- American Red Cross (2015) “Drones for disaster response and relief operations,” (April), p. 51.
- Tkach, I. and Edan, Y. (2019) Distributed heterogeneous multi sensor task allocation systems. Springer International Publishing (Automation, Collaboration, & E-Services).
- Chapman, A., Micillo, R., Kota, R. and Jennings, N. (2009) “Decentralised dynamic task allocation: a practical game-theoretic approach,” in, pp. 915–922. <https://doi.org/10.1145/1558109.1558139>.
- Al-Buraiki, O. and Payeur, P. (2019) “Probabilistic Task Assignment for Specialized Multi-Agent Robotic Systems,” in, pp. 1–7. <https://doi.org/10.1109/ROSE.2019.8790420>.
- Robin, C. and Lacroix, S. (2015) A taxonomy of multi-robot target detection and tracking problems.
- Jang, I. (2018) Effective task allocation frameworks for large-scale multiple agent systems.
- Bhattacharya, S., Ghrist, R., Kumar, V.: Multi-robot coverage and exploration on Riemannian manifolds with boundaries. *Int. J. Robot. Res.* **33**, 113–137 (2014). <https://doi.org/10.1177/0278364913507324>
- Kolling, A., Carpin, S.: Pursuit-evasion on trees by robot teams. *IEEE Trans. Rob.* **26**(1), 32–47 (2010). <https://doi.org/10.1109/TRO.2009.2035737>
- Moors, M., Rohling, T. and Schulz, D. (2005) A probabilistic approach to coordinated multi-robot indoor surveillance,” in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3447–3452
- Yu, H., Beard, R. W., Argyle, M. and Chamberlain, C. (2011) Probabilistic path planning for cooperative target tracking using aerial and ground vehicles,” in Proceedings of the 2011 American Control Conference, pp. 4673–4678
- Espinós Longa, M., Inalhan, G. and Tsourdos, A. (2022) Swarm Intelligence in Cooperative Environments: Introducing the N-Step Dynamic Tree Search Algorithm,” in AIAA SCITECH 2022 Forum. Reston, Virginia: American Institute of Aeronautics and Astronautics, pp. 1–13. <https://doi.org/10.2514/6.2022-1839>.
- Espinós Longa, M., Tsourdos, A. and Inalhan, G. (2022) Swarm intelligence in cooperative environments: N-Step dynamic tree search algorithm extended analysis. *American Control Conference* (accepted), pp. 1–13.
- Espinós Longa, M., Inalhan, G. and Tsourdos, A. (2022) “Swarm intelligence in cooperative environments: N-Step dynamic tree search algorithm overview. *J. Aerospace Inform. Systems* (submitted).
- Singh, P., Dhiman, G.: A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches. *Journal of Computational Science* **27**, 370–385 (2018). <https://doi.org/10.1016/j.jocs.2018.05.008>
- Tripathy, M., Mishra, S.: Bacteria foraging-based solution to optimize both real power loss and voltage stability limit. *IEEE Trans. Power Syst.* **22**(1), 240–248 (2007)
- Bayindir, L. (2016) A review of swarm robotics tasks. *Neurocomputing*, 172(August 2015), pp. 292–321. <https://doi.org/10.1016/j.neucom.2015.05.116>.
- Ducatelle, F., Förster, A., Di Caro, G. and Gambardella, L. M. (2009) New task allocation methods for robotic swarms, in.
- Pini, G., Brutschy, A., Birattari, M. and Dorigo, M. (2009) “Interference reduction through task partitioning in a robotic swarm,” ICINCO 2009 - 6th International Conference on

- Informatics in Control, Automation and Robotics, Proceedings, 2 RA, pp. 52–59. <https://doi.org/10.5220/0002195200520059>.
22. Huang, L., Ding, Y., Zhou, M., Jin, Y., Hao, K.: Multiple-Solution Optimization Strategy for Multirobot Task Allocation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **50**(11), 4283–4294 (2020). <https://doi.org/10.1109/TSMC.2018.2847608>
 23. Abielmona, R., Falcon, R., Zincir-Heywood, N. and Abbass, H. (2016) Recent advances in computational intelligence in defense and security. https://doi.org/10.1007/978-3-319-26450-9_1.
 24. Pandian, A. (2013) Training neural networks with ant colony optimization.
 25. Mavrovouniotis, M. and Yang, S. (2013) Evolving neural networks using ant colony optimization with pheromone trail limits, in 2013 13th UK Workshop on Computational Intelligence (UKCI), pp. 16–23.
 26. Blum, C. and Socha, K. (2005) “Training feed-forward neural networks with ant colony optimization: An application to pattern classification, in Fifth International Conference on Hybrid Intelligent Systems, HIS 2005, pp. 6 pp.-. <https://doi.org/10.1109/ICHIS.2005.104>.
 27. Li, Q., Jiang, Z.-P.: Two decentralized heading consensus algorithms for nonlinear multi-agent systems. *Asian Journal of Control* **10**(2), 187–200 (2008). <https://doi.org/10.1002/asjc.018>
 28. Wang, J., Sun, Y., Zhang, Z., Gao, S.: Solving multitrip pickup and delivery problem with time windows and manpower planning using multiobjective algorithms. *IEEE/CAA Journal of Automatica Sinica* **7**(4), 1134–1153 (2020). <https://doi.org/10.1109/JAS.2020.1003204>
 29. Hu, B., Cao, Z., Zhou, M.: An efficient RRT-based framework for planning short and smooth wheeled robot motion under kinodynamic constraints. *IEEE Trans. Industr. Electron.* **68**(4), 3292–3302 (2021). <https://doi.org/10.1109/TIE.2020.2978701>
 30. Jian, Y.-L., Lian, F.-L. and Lee, H.-T. (2008) Deployment of a team of biomimetic searching agents based on limited communication quantity. *Asian J. Control*, **10**(4). <https://doi.org/10.1002/asjc.043>.
 31. Li, P. and Duan, H. (2014) Bio-inspired computation in unmanned aerial vehicles, in *Bio-Inspired Computation in Unmanned Aerial Vehicles*, pp. 35–69. https://doi.org/10.1007/978-3-642-41196-0_2.
 32. Girma, A., Bahadori, N., Sarkar, M., Tadewos, T.G., Behnia, M.R., Mahmoud, M.N., Karimodini, A., Homaifar, A.: IoT-enabled autonomous system collaboration for disaster-area management. *IEEE/CAA Journal of Automatica Sinica* **7**(5), 1249–1262 (2020). <https://doi.org/10.1109/JAS.2020.1003291>
 33. Corne, D., Reynolds, A. and Bonabeau, E. (2012) “Swarm Intelligence,” in *Handbook of Natural Computing*, pp. 1599–1623.
 34. Nedic, N., Prsic, D., Dubonjic, L., Stojanovic, V., Djordjevic, V.: “Optimal cascade hydraulic control for a parallel robot platform by PSO”, *The International Journal of Advanced Manufacturing Technology* 2014 72:5. *Springer* **72**(5), 1085–1098 (2014). <https://doi.org/10.1007/S00170-014-5735-5>
 35. Potter, M. A. and de Jong, K. A.: A cooperative coevolutionary approach to function optimization, in Davidor, Y., Schwefel, H.-P., and Männer, R. (eds) *Parallel Problem Solving from Nature*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 249–257. (1994)https://doi.org/10.1007/3-540-58484-6_269.
 36. Panait, L., Wiegand, R. P. and Luke, S. (no date aj) Improving Coevolutionary Search for Optimal Multiagent Behaviors.
 37. Ficici, S. G. and Pollack, J. B. (2000) “A game-theoretic approach to the simple coevolutionary algorithm,” in Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H.-P. (eds) *Parallel Problem Solving from Nature*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 467–476. https://doi.org/10.1007/3-540-45356-3_46.
 38. Haynes, T., Wainwright, R., Sen, S. and Schoenefeld, D. (1995) Strongly typed genetic programming in evolving cooperation strategies.,” in *Proceedings of the 6th International Conference on Genetic Algorithm*, pp. 271–278.
 39. Passino, K. M. (2002) Biomimicry of bacterial foraging, *IEEE Control Systems Magazine*, (June), pp. 52–67.
 40. Quijano, N. and Passino, K. M. (2007) Honey bee social foraging algorithms for resource allocation: theory and application, pp. 1–39.
 41. Cabreira, T., Brisolaro, L., Ferreira, P.R., Jr.: Survey on coverage path planning with unmanned aerial vehicles. *Drones* (2019). <https://doi.org/10.3390/drones3010004>
 42. Liu, Z.-N., Liu, X.-Q., Yang, L.-J., Leo, D. and Zhao, H.-W. (2018) An autonomous dock and battery swapping system for multirotor UAV, (May). <https://doi.org/10.13140/RG.2.2.19437.90085>.
 43. Rohan, A., Rabah, M., Asghar, F., Talha, M., Kim, S.-H.: Advanced drone battery charging system. *Journal of Electrical Engineering & Technology* **14**(3), 1395–1405 (2019). <https://doi.org/10.1007/s42835-019-00119-8>
 44. Valente, J., Sanz, D., del Cerro, J., Barrientos, A., de Frutos, M.Á.: Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields. *Precision Agric.* **14**(1), 115–132 (2013)
 45. Artemenko, O., Dominic, O. J., Andryeyev, O. and Mitscheltiel, A. (2016) Energy-aware trajectory planning for the localization of mobile devices using an unmanned aerial vehicle, 2016 25th International Conference on Computer Communication and Networks (ICCCN), pp. 1–9.
 46. Xu, A., Viriyasuthee, C. and Rekleitis, I. (2011) Optimal complete terrain coverage using an unmanned aerial vehicle. *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2513–2519. <https://doi.org/10.1109/ICRA.2011.5979707>.
 47. Öst, G. (2012) Search path generation with UAV applications using approximate convex decomposition. Linköping University, The Institute of Technology.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Marc Espinós Longa completed his BSc in Aerospace Engineering at Polytechnic University of Catalonia (UPC) between 2014 and 2018 and achieved a Double Master's Degree in Aeronautics and Autonomous Vehicles at Polytechnic University of Catalonia and Cranfield University from 2018 to 2020, respectively. During that period of time, he led the full development of a semi-autonomous swarm of UAV at BAE UAV Swarm Challenge 2019, became lecturer in statistic and probability at ASES Academia and completed his individual research project in swarm intelligence with BAE Systems. Marc is currently a PhD Researcher in Aerospace Engineering at Cranfield University sponsored by BAE Systems and Engineering and Physical Sciences Research Council. His main area of research is artificial intelligence applied to goal-based swarming in cooperative environments.

Professor Antonios Tsourdos is Head of the Center of Autonomous and Cyber-Physical Systems and Director of Research – SATM. He is Chair of the IFAC Technical Committee of Aerospace and editorial member for a number of journals including IEEE Transaction of Aerospace & Electronics Systems, Aerospace Science and Technology, IMechE Journal of Aerospace Engineering, Control Engineering Practice and International Journal of Systems Science. He has been involved in a number of projects including EPSRC, H2020, IUK, Sesar and Clean Sky funded projects focusing on flight envelope protection,

data analytics, system architectures, urban air mobility, communication and networking, cyber-physical aerospace systems and avionics. He has published widely and has been recipient of the UK MoD Autonomy Grand Challenge award and the IET Team Innovation award.

Gokhan Inalhan received his BSc degree in Aeronautical Engineering from Istanbul Technical University in 1997, and MSc and PhD degrees in Aeronautics and Astronautics from Stanford University in 1998 and 2004, respectively. In 2003, he received a PhD Minor from Stanford University in Engineering Economics and Operations Research (currently, Management Science and Engineering). Between 2004 and 2006, he worked as a Postdoctoral Associate at Massachusetts Institute of Technology. During this period, he led the Communication and Navigation group in the MIT-Draper Laboratory NASA CER project. He has served as the Director of the Controls and Avionics Laboratory (2016 – 2019) at Istanbul Technical University. Gokhan is currently the BAE Systems Chai, Professor of Autonomous Systems and Artificial Intelligence and Deputy Head of the Center for Autonomous and Cyber-Physical Systems at Cranfield University. He and his research group focus on design, modeling, GNC, resilience and security aspects of autonomy and artificial intelligence for air, defense, transportation and space systems.