Applied Soft Computing Journal 89 (2020) 106135

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

An enhanced decentralized artificial immune-based strategy formulation algorithm for swarms of autonomous vehicles

Marios Stogiannos [a,b], Alex Alexandridis [a,*], Haralambos Sarimveis [b]

[a] Department of Electrical and Electronic Engineering, University of West Attica, Thivon 250, Aigaleo 12241, Greece
[b] School of Chemical Engineering, National Technical University of Athens, Iroon Polytechneiou 9, Zografou 15780, Athens, Greece

ARTICLE INFO

Article history:
Received 13 May 2019
Received in revised form 13 January 2020
Accepted 24 January 2020
Available online 28 January 2020

Keywords:
Artificial immune system
Autonomous vehicle swarm
Decentralized path planning
Optimal task allocation
Swarm intelligence

ABSTRACT

This work presents an algorithmic approach to the problem of strategy assignment to the members of a swarm of autonomous vehicles. The proposed methodology draws inspiration from the artificial immune system (AIS), where a large number of antibodies cooperate in order to protect an organism from foreign threats by local exchange of information. The decentralized nature of the methodology does not suffer from problems like the need of a central control unit, the high maintenance costs and the risks associated with having a single point of system failure, which are common to centralized control techniques. Decentralized and distributed optimization schemes employ simple algorithms, which are fast, robust and can run locally on an autonomous unit due to their low processing power requirements. In contrast to standard AIS-based decentralized schemes, the proposed methodology makes use of a dynamic formulation of the available strategies and avoids the possibility of choosing an invalid strategy, which may lead to inferior swarm performance. The methodology is further enhanced by a dual strategy activation decay technique and a blind threat-follow rule. Statistical testing on different case studies based on "enemy search and engage" type scenarios in a simulated environment demonstrates the superior performance of the proposed algorithm against the standard AIS, an enhanced AIS version and a centralized particle swarm optimization (PSO) based methodology.

1. Introduction

As technology advances, computational systems are continuously becoming more complex in order to transform the vast amounts of raw data into meaningful information that may assist in producing sensible answers to contemporary scientific questions, or even steer researchers towards new unexplored paths. The problem which becomes more and more evident is that the amount of raw data increases significantly faster than the processing capabilities of information systems. A solution came by distributed and decentralized systems [1], where data are partitioned and processed in individual clusters. The advantages made apparent by parallel processing of data clusters find use in applications like swarm robotics [2,3], where multiple systems smaller in size and less energy-demanding, are usually preferable compared to a single powerful system.

Any field utilizing robot swarms can benefit from a distributed and decentralized control system, where, as of now, centralized controllers [4] were the industry standard approach, due to specific advantages, namely (a) easy implementation and (b) improved results when compared to decentralized schemes, as the former make use of information from all swarm members, whereas the latter rely on local information only. On the other hand, the disadvantages of centralized controllers are mainly: (a) very high computational cost especially for large swarms, because of the exponential increase in processing power demands as the swarm size increases, (b) the need for reliable and secure communications between the main processing unit and all the members of the swarm in order for an unobstructed information exchange to take place and (c) the increased cost of back-up processing and power-generating systems, in case the main system fails, or goes offline. The most important drawback of a centralized system is that the main processing unit itself may become a single point of catastrophic failure for the whole system, either due to an accident, or to an external attacker.

Decentralized algorithms [5,6] are usually simple enough to run locally for each member of the swarm in real-time, without requiring high processing power, or any specialized hardware. Numerous applications so far have proven the reliability and scalability of decentralized approaches, as well as their adaptability to solving different kinds of problems. In a recent work [7], the tracking control problem for interconnected stochastic nonlinear systems is solved by a decentralized control method. The methodology is applied to a system of two interconnected inverted pendulums on carts and the errors are shown to converge

[*] Corresponding author.
E-mail address: alexx@uniwa.gr (A. Alexandridis).

https://doi.org/10.1016/j.asoc.2020.106135
1568-4946/© 2020 Elsevier B.V. All rights reserved.

to a desired value. Standard nonlinear model predictive control has also been modified to accommodate decentralized applications as in [8], where the proposed approach is shown to be effective on the problem of controlling a large power system under a wide range of operating conditions. The problem of assigning the optimal charging strategy to electric vehicles is tackled through a decentralized methodology based on the augmented Lagrangian method and the alternating direction multiplier method in [9], where it is shown that the revenue of an electric vehicle aggregator can be further improved, while increased computational cost and communication privacy issues which are inherent in centralized systems are avoided. Computational intelligence techniques, like neural networks, have also been coupled with decentralized methodologies to formulate control schemes, for instance in [10,11], where the problems of stabilization of nonlinear multi-agent systems and the cooperative search of a swarm of unmanned aerial vehicles are addressed, respectively. An online decentralized approach to swarm path planning considering obstacle avoidance is presented in [12], while in [13] the same problem is considered by integrating a receding horizon technique to a decentralized approach.

Bio-inspired methods mostly used to synthesize centralized control schemes have also been widely employed in the search for optimality through information distribution and decentralization. Recent examples of such formulations include the design of a load-frequency controller for interconnected power systems [14] and the frequency–voltage control of a wind turbine-load-battery plant [15], both of which use the well-known particle swarm optimization (PSO) algorithm. The ant colony optimization algorithm is another example of a bio-inspired method modified to form a decentralized controller [16], where autonomous robots individually select the optimal task in a robust and efficient way.

Artificial immune systems (AIS), also employed in this work, have been widely used in decentralized control systems, as well as in a number of other applications like clustering and classification [17–19], music generation [20], email spam detection [21], intrusion detection in computer systems [22,23], etc. Due to their distributed nature, AIS have been evolved specifically for solving decentralization and optimal decision-making problems. AIS-based methodologies stem from the mechanism of the biological immune system (BIS), which tries to solve the complex problem of defending an organism against natural threats by distributing information to a swarm consisting of a very large number of agents.

The mobile robot navigation problem is solved in [24] by successfully introducing AIS techniques to a genetic algorithm approach, but the presented methodology requires an initial behavior parameter calculation via simulations of possible scenarios that needs up to 25 min before deployment of a real robot. The main advantage is the high adaptability and robustness of the resulting robot behavior in closed space environments. In [25] an AIS-based generic control methodology is developed and applied to the coordination of autonomous agents within an intelligent transport system framework. The proposed scheme has a superior performance, but the complexity of the required behaviors may be an issue when applied to scenarios with simpler tasks, like surveillance or search and rescue missions. A three level AIS-inspired framework has been developed in [26] for heterogeneous mobile robots performing various tasks with promising results, but the proposed methodology is bound by the need of central control. In [27–29] a swarm of robots is deployed to search and carry out abstract tasks, while also evaluating the task density of an area, so as to decide whether to call for aid or not. The presented methodologies optimally aggregate the swarm around the target, but such designs are not very effective in cases where one agent may be sufficient for a given task and where flocking should be avoided to increase the exploration capabilities of the swarm. The case of cooperative cargo transport is tested in [30] through an AIS-based immune agent network methodology, which showed notable results. Even though the original tuning was made by hand, a reinforcement learning technique allowed for adaptation to the unknown task parameters, but the original strategy pool from which the robots pick their actions is not adaptive. In [31] the trajectory tracking control problem is addressed for a swarm of spacecrafts under interferences and uncertainties in an outer space scenario. This approach integrates ideas based on AIS within the overall control framework in order to minimize the dependency of the response to the models of the system and to the unknown environment disturbances, but as the previously mentioned work, it employs a fixed and non-adaptive AIS strategy pool. Unmanned aerial vehicle coordination is evaluated in an area surveillance scenario in [32], where a swarm is tasked with the surveillance and defense of an area. The presented methodology obtains good performance in terms of computational burden, expandability and robustness, while also keeping flocking to a minimum and not requiring parameter tuning in the case of changes to the initial swarm configuration. This method, which also employs a fixed strategy pool, is discussed extensively in subsequent sections and also used for comparison purposes to show the performance gained from using an adaptive strategy pool.

This work aspires to fill research gaps in the literature employing AIS-based methodologies, which have been identified as follows:

- The number of available strategies for each swarm member is fixed and predetermined before deployment. The problem arising from a standard pool of strategies is that the agents are not able to identify a threat in an explicit manner, but instead, they can only choose to enter a mode where they detect a target and perform the strategy's predefined action, which may not fit the circumstances at hand. A variable strategy pool has the required flexibility to dynamically adapt to multiple different threats.
- The strategy selection is made by a process utilizing the information collected by each agent's local environment. In this line of thought, AIS-based methodologies commonly assume that the local environment contains enough information for an optimal strategy selection, but as will be further explained later, this in not always the case, ultimately leading to a number of agents not being successfully exploited.
- AIS-based decentralized methodologies have not been developed to the point where they may approximate or match the optimality level of their centralized counterparts. The most usual argument in favor of the decentralized algorithms (in some cases being the only one) is the computational cost, which is always very low compared to centralized approaches.

The main novelty of this work lies in the enhancement of the standard AIS algorithm, which overcomes the disadvantage of having to choose among a fixed pool of available strategies, while efficiently addressing all other objectives. More specifically, the contributions of the proposed AIS-based methodology are the following:

- A pool of strategies is dynamically formulated at each discrete time instant according to the previously and currently detected agents and intruders. The methodology identifies each nearby friend or foe, leading to a variable size strategy pool and adding to the versatility of each agent's choice of action.

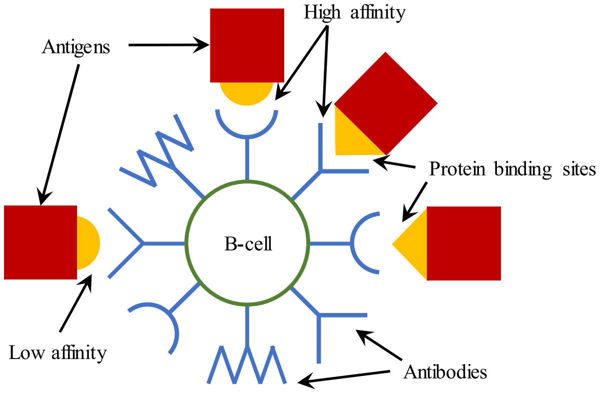


Fig. 1. Interactions between antibodies of a B-cell with free-roaming antigens.

- The validity of past strategies is determined based on the existence of previously assigned targets at the current time step. Weakening of invalid strategies takes place, ultimately leading to the strengthening of the best possible choice.
- If an agent cannot keep up with its target, instead of choosing a random path to follow, the methodology takes advantage of historical data, to increase the probability of obtaining a path closer to the target's unknown position.

The direct benefit stemming from the collective merits of this approach is better area exploration without compromising the performance in engaging threats, since the swarm members are efficiently allocated to feasible tasks. The robustness of the methodology is further developed by enhancing the technique used to reject invalid strategies, in order to exploit the last known positions of previously found threats.

The rest of the paper is organized as follows. Section 2 includes an introduction to the biological immune system (BIS), which is the main inspiration for this work. The initial modeling of the BIS, leading to the development of AIS, is also explained in brief. Section 3 describes the problem of area surveillance using an unmanned aerial vehicle swarm, in which the AIS is applied in order for each member to decide the best strategy, so that the whole swarm performs optimally under certain restrictions. Section 4 details the algorithm employed by the standard AIS methodology, while also describing an enhanced AIS approach, both of which are used in the test cases for comparison purposes. Section 5 describes in detail the proposed methodology. In the next section a number of tests are designed to cover a wide range of possible cases for the given scenario and the corresponding statistical significance test results are discussed, including comparisons with different decentralized and centralized schemes. Section 7 concludes this paper by summarizing the results and suggesting directions for future research. The included appendix provides a more detailed analysis of the case studies.

2. The immune system

2.1. Biological immune system (BIS)

Every living organism has developed a system to protect itself against disease, called an immune system [33]. In mammals, leukocytes (white blood cells) interact with foreign substances and carry different tasks from cleaning cellular debris to identifying and eliminating diseases. Antigens are the organisms which invade the body of the host.

The B-cells (a type of leukocytes) identify a threat by trying to bind the antibodies on their surface to the antigens found. The identification boils down to a protein binding process between the antibody and the antigen with a success rate, called *affinity*. Fig. 1 qualitatively depicts the possible interactions between an antibody-carrying B-cell and a number of different types of antigens. The antibody that achieves a high affinity is cloned to a very large number and the clones are released to the bloodstream to engage the identified antigens. The cloning process facilitates small mutations resulting to greater affinity scores; however, before being released, the antibodies undergo a self-nonself discrimination procedure, which certifies their ability to distinguish the organism's own cells. The antibodies spread and bind to any same antigen found, effectively marking it for destruction by T-cells or NK-cells. After the threat is negated, a small number of the antibodies with the highest affinity remain active in the bloodstream to increase the response to the same threat in the future, creating an immunological memory to a specific antigen. This cloning and mutation process gives the immune system the capability to evolve and adapt to newly encountered threats.

2.2. Idiotypic network

Niels Jerne pioneered in solving the most intriguing problems in the field of immunology almost 35 years ago, winning the Nobel prize for his classical work [34]. Jerne constructed a differential equation [35], which fully describes the dynamics between lymphocytes. In the same work he suggested that the immune system cells do not work independently, but rather each cell interacts with its neighboring cell in order for both to mutually suppress or stimulate each other.

The outcome of this interaction is that each cell calculates an activation value for the antibodies that it carries. The most suitable antibody attains the highest activation score and is selected to be cloned in order for the number of suitable antibodies to become large enough so that the threat may be successfully fought [36].

2.3. Artificial Immune System (AIS)

Farmer used Jerne's idiotypic network theory to form mathematical models [37,38] describing the biological immune system's mutual lymphocyte interaction with the purpose of determining the most suitable antibody. In order to do that, the concentration x of each antibody $i = \begin{bmatrix} 1 & 2 & \dots & N \end{bmatrix}$ needs to be specified, N being the total number of existing antibody types:

$$\dot{x}_i = c\left[\sum_{j=1}^{N} m_{j,i}x_i x_j - k_1 \sum_{j=1}^{N} m_{i,j}x_i x_j + \sum_{j=1}^{n} m_{j,i}x_i y_j\right] - k_2 x_i \quad (1)$$

where c is the rate between antibody-to-antibody and antibody-to-antigen interaction and can be viewed as a measure of the rate of antibody stimulations, suppressions and clonings, which happen in the course of time. Mutual stimulation between two antibodies, one of type i and another of type j, where $1 \le i, j \le N$, is represented by the first term of Eq. (1), while m_{ji} is the mutual stimulation coefficient between these antibody types, which will be explained later in detail. The second term represents the mutual suppression between antibody types, where $m_{i,j}$ is the mutual suppression coefficient and k_1 may be used to induce a predisposition between stimulation and suppression. The third term represents the interaction between the antigens and the antibodies, where y_j is used to model the antigen concentration changing over the course of the immune system's response to it. The number of identified antigen types is denoted as n, while m_{ji} defines the binding affinity between antibody and antigen types. The last term represents the decay rate at which antibodies die when the infection is overcome. Obviously, the decay rate coefficient k_2 should take on a value which will not aggressively destroy antibodies while an infection is still at large.

3. Problem statement

The group of methodologies inspired by the standard form of the AIS display a number of very useful advantages when applied to problems of decentralized decision-making between members of a group with a mutual goal [39–41]. AIS-based control schemes inherit the merits of their biological analogue and, thus, are very effective in applications where the decision to aggregate or disperse a swarm of agents is of critical importance [32]. In such a framework, where a balance between the opposing goals of space exploration and exploitation must be struck, the AIS is a popular choice.

3.1. Simulation environment

In this work, an environment as the one described earlier is devised to allow for the simulation and comparison of several test cases as described below. A two-dimensional square map is set to facilitate a battle scenario. A swarm of point-mass UAVs, called agents, are deployed for surveillance and protection of a specified area. At the same time, another set of UAVs, called intruders, appear within the same area. The goal of the agents is to follow the intruders after detection, while certain limitations exist. Actions that take place after establishing contact and deciding to follow an intruder (such as capturing, immobilization, or destruction strategies) are not considered in this study and are left to be independently implemented by each application designer. Trajectory modeling is not studied in this work, but all of the UAVs have a preset max speed and turning angle limits, while motion is performed in straight lines between timesteps. Simple as it is, this type of motion modeling is realistic enough for our conclusions to be clear and robust. In a real-life scenario, the agents would carry electronics for the purpose of identifying intruders' position and communicating important data with other agents, both of which are within a specific range. Again, the simplest case of immediately determining the exact position is assumed, while the communications between agents are considered disturbance-free. Note that intruder detection and agent communications radii are not the same in the simulations.

3.2. Initial strategies and decision making

At each simulation timestep, every agent decides between a number of strategies which boil down to three main modes of operation. The first one is the "patrol" mode (denoted by "P"), which is performed by following a random trajectory within the map area. The second mode is to "engage an intruder" (denoted by "D"), where an agent actively follows the trajectory of a nearby detected intruder. The third one is "follow an agent" (denoted by "F"), which is a special case chosen when another agent follows an intruder, communicating this information to nearby agents who may decide to come to its aid.

To make the actual decision, all members of the swarm calculate the activation of each strategy according to their own data, coming strictly from their local environment. Such data include: (1) the strategy activations carried by nearby detected agents, (2) the distances of nearby detected agents and (3) the distances of nearby detected intruders. It is obvious that these criteria may be expanded, modified or redefined in order to adapt this methodology to a specific application. In the end, the strongest strategy is selected to be executed at each time step.

At this point, an analogy between the biological immune system and the previous battle scenario needs to be made, in order for the relationships to become clear. Each agent carries a number of strategies which is analogous to a B-cell carrying a number of antibodies, while the intruders are related to the antigens. The stimulation or suppression of the antibodies is directly related to the calculation of the strategy activations which is based on information gathered from nearby agents.

Table 1
Mutual interaction coefficient $m_{i,j}$ values for SS and TE-SS AIS methodologies

	P	D	F$_p$
P	+1.0	−0.8	−0.4
D	−0.4	+1.0	−0.4
F$_q$	−0.6	−0.4	+1.0, if $p = q$ −0.4, if $p \neq q$

4. Standard and thymus-enhanced AIS methodologies

The algorithmic approach for the standard AIS, applied on the strategy assignment problem of UAV swarms, consists of the following steps, which are executed at each sample time by each agent:

1. Sweep environment to detect agent and intruder positions within sensor range.
2. Communicate with nearby agents to obtain their strategy strengths.
3. Calculate own strategy activations.
4. Execute the strategy with the highest activation value.

The first two steps are performed by the UAV's onboard electronics. There is a number of methods for obtaining an object location in 3D-space [42], as well as establishing communication for data transfer between moving vehicles in scientific literature [43], but these two steps are out of the scope of this work. To perform the third step, a simplified discrete form of Eq. (1) has been used in many works [27–30,32]. The most common form to calculate the current strength $S_i^k(t)$ of strategy i for agent k comprises three components:

$$S_i^k(t) = Sa_i^k(t) + Sb_i^k(t) + Sc_i^k(t) \qquad (2)$$

where $Sa_i^k(t)$ is the mutual interaction between any pair of agents, $Sb_i^k(t)$ represents the intruder effect upon a specific strategy and Sc_i^k is a term controlling the weakening of a strategy. The mutual interaction $Sa_i^k(t)$ between two agents represents the strategy stimulation or suppression based on the actions taken by nearby agents and is given by:

$$Sa_i^k(t) = \frac{a}{N}\sum_j^N \sum_{l\in Ic^k(t)} m_{i,j} c_j^l(t-1)\, c_i^k(t-1) \qquad (3)$$

where $Ic^k(t)$ is the set of agents within communication range of agent k at timestep t. The coefficient a represents the weight of the agent-to-agent interaction on the total strategy strength and is a designer-defined value which can be used as a tuning parameter. The concentration c_i^k of strategy i for agent k is obtained by expressing the appropriate strategy strength in a percentage of all the N strategy strengths carried by the corresponding agent at a specific time. The mutual interaction weighting coefficient $m_{i,j}$ between strategies i and j for any pair of agents is defined in Table 1 and is used to show the effect of the strategy executed by another agent on the strategy strength i for agent k. The p and q indices used in Table 1 are meant to define the actual id of the agents to be followed when these strategies are selected. To clarify this point, let us assume agent 1 ($k = 1$) is currently calculating the mutual interaction $Sa_i^k(t)$ for a strategy F_3 ($q = 3$), so that the agent will attend to a request to follow agent 3, should they decide to choose this strategy. Agent 2 ($l = 2$) performs strategy F_5 ($p = 5$) meaning that it currently follows agent 5. In this case $p \neq q$ and $m_{i,j} = -0.4$. If $p = q$, then the mutual interaction coefficient becomes $m_{i,j} = 1.0$. These values are obviously chosen as they facilitate flocking around an intruder and, thus, the swarm responds more like the actual BIS. The intruder effect $Sb_i^k(t)$ on strategy i of agent k is defined by the sum of environmental impacts $\varphi^{k,m}$ of the set of detected intruders $Id^{k}(t)$ acting upon the previous concentration value of the same strategy and weighted by the term coefficient b.

$$Sb_{i}^{k}(t) = b\sum_{m\in Id^{k}(t)}\varphi^{k,m}c_{i}^{k}(t-1) \tag{4}$$

The environmental impact $\varphi^{k,m}$ is an abstract notion, which may describe any number of application-dependent relationships between an agent k and an intruder m, for example the ability to engage in battle, the probability of winning against a foe, or the ability to follow an intruder through specific environmental conditions. In our study, we use the agent–intruder proximity as a metric for the preference that an agent will show to a specific intruder, which is common to many similar applications [44]. This method will effectively lean towards choosing the closest intruder to follow, such that

$$\varphi^{k,m}(t) = 1 - \frac{d^{k,m}(t)}{r_{\mathrm{d}}} \tag{5}$$

where $d^{k,m}(t)$ is the agent–intruder Euclidean distance and r_{d} is the agent's detection range.

The last component Sc_{i}^{k} is called decay rate and models the rate of strategy i weakening over time, if it is not stimulated by an external factor.

$$Sc_{i}^{k}(t) = -\mu c_{i}^{k}(t-1) \tag{6}$$

where the decay rate coefficient μ may be a simple percentage of the previous concentration value. The biological analogue of this component is the reduction of antibody count after an infection is subdued; when this happens, there is no need for the vast number of cloned antibodies to remain active in the bloodstream. The final activation value is determined by adding the newly calculated strength, weighted by a sigmoid function, to the previous activation value.

$$A_{i}^{k}(t) = A_{i}^{k}(t-1) + \frac{S_{i}^{k}(t)}{1+e^{\sigma - A_{i}^{k}(t-1)}}\cdot\Delta t \tag{7}$$

where a tuning parameter σ controls the current strategy strength suppression.

In 2014, Weng et al. [32] introduced a modification which improves the performance of the standard AIS algorithm, by tackling the problem caused by the possibility of knowledge explosion. This problem may slow down the responsiveness of the system, by not letting individual agents adapt quickly to current changes in their environment. The modification includes a critic, which evaluates whether the number of neighboring agents that perform the same strategy has reached a predefined threshold C_{th}, in which case the second strongest strategy is selected. This technique is capable of avoiding the possibility where an agent overloaded with heavily diverse data is forced to follow the wrong strategy and ultimately miss a correct and obvious choice. Results included in the publication, show that the modification performs better than the standard AIS in all cases tested, even in a case where all agents exchange data with all other agents. The swarm behavior improvement is indeed notable, but the rate of success drops significantly as the effect of information overload increases.

5. Proposed variable-strategy AIS methodology

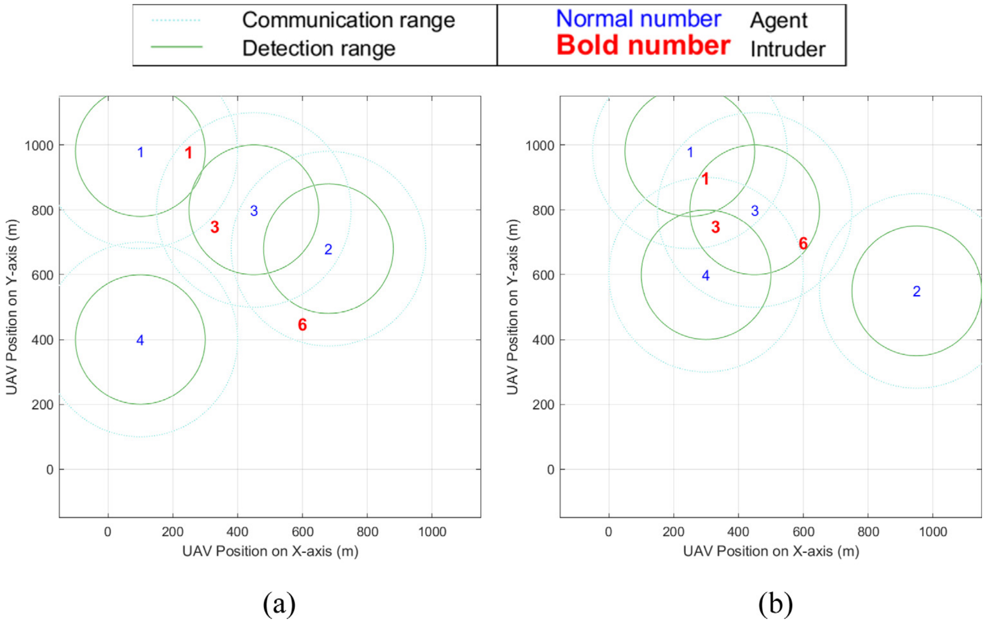
In this work, a different approach to the strategy-driven swarm coordination problem is considered. The standard AIS model described in Section 4 is regarded and expanded to include the feature of a variable number of valid strategies. In most AIS-based works, the strength of a strategy varies according to each agent's environment; however, the number of strategies is always fixed and the most fit of them is selected to be executed.

It is a fact that in such approaches the originally included strategies cannot adapt to the future environment an agent may find itself in. The problem appears in cases where a strategy is not at all relevant at a specific time step, but amounts to the greatest strength, nonetheless. This may come as a result of the simultaneous effect of past strategy strength accumulation, along with an insufficient strength suppression methodology, i.e. a small decay rate. The practical outcome may be a case where an agent selects to follow an intruder, while no intruder exists in their detection range. That may come as a result of the strategy to follow an intruder being strengthened by the exchange of information with nearby agents to an amount that the inherent strategy strength decay ability is not able to cope with. This effect may not be entirely unwanted and may even be exploited as will be described later, but if a compensating technique does not exist, then the agent is left with a strategy to follow a target, but without having a specific target to follow. If such a strategy remains active for a long time and/or by a large part of the swarm, it may eventually lead to the swarm failing to realize the predefined synergistic goal, which in our case is the area defense scenario. Even techniques that avoid knowledge explosion, like the one described in the previous section, are prone to an extent to this effect.

To address this problem, a pool of valid strategies based on available and historic data is formulated and then the agent calculates the new strategy strengths considering past values. This pool is updated at each time instant, while emerging conditions give rise to new strategies, or invalidate older ones based on designer-defined rules.

The real-world UAV parameters (positioning, speed and turning angle limitations, agent tagging, area coverage, etc.) and the tuning parameters of the algorithm are selected off-line before the swarm comes into operation. The strategy pool of every agent is initialized with a single patrol (P type) strategy, which is assumed to be the only available strategy. The initial activation and concentration parameters of the strategy are set to zero (0) and one (1), respectively. No other initialization step is required.

In order to update the pool, at each time instant, the agent searches for other agents, as well as intruders, and assigns a unique tag to each new one, while the old ones retain their previous tags. Fig. 2 depicts an example of strategy formulation for a random agent, where at a certain point in time $t = 0$, an agent's (agent 3) strategy pool includes the P strategy, a second strategy tagged D_{I3} (short for "engage intruder 3") and a third one tagged F_{A2} (short for "follow agent 2"). At the next time instant $t = 1$, two intruders (I1 and I6) and two agents (A1 and A4) appear and are discovered, while agent 2 moves out of communication range. Thus, the updated pool will include P (the patrol strategy is always included in the pool by default), along with D_{I1}, D_{I3}, D_{I6}, F_{A1}, F_{A2}, and F_{A4}. A strategy for following agent 2 remains, even though it is now outside the communication range. This strategy contains information that may give rise to a significant advantage as will be further explained later. The same logic would be applied in the case of a detected intruder that moves out of the detection range. Hence, the formulated pool would include one strategy for patrol, three strategies where an intruder would be engaged and three more strategies where an agent would be followed. For each of these strategies, a strength would be calculated taking into account the information exchanged with the two agents in communications range (A1 and A4) only, as well as other conditions, i.e. distances, etc. as in the standard methodology. The difference here would be that each agent would communicate the strengths only for the strategies they share, unlike the standard methodology, where all strength



(a)

(b)

Fig. 2. Strategy pool formulation example for agent 3 at time instants (a) $t = 0$, and (b) $t = 1$.

values are shared. The strength of each strategy is still calculated by the sum of three components as shown in Eq. (2), but now the first part, corresponding to Eq. (3), becomes

$$Sa_i^k(t) = \begin{cases} 0, & I_k^L(t) = 0 \\ \dfrac{a}{I_k^L(t)} \displaystyle\sum_{l=1}^{I_k^L(t)} maa_{i,l} c_i^l(t-1) c_i^k(t-1), & I_k^L(t) > 0 \end{cases} \quad (8)$$

where $I_k^L(t)$ is the number of all other agents within communication range of agent k who share the same strategy and, thus, whose communicated data participates in the calculation of Eq. (8). The agent-to-agent interaction weighting coefficient $maa_{i,l}$ takes into account two facts, the first one being strategy i and the second one being whether agent l currently executes the same strategy i, or a different one. At this point, it must be clarified that agent l will always carry strategy i in order to qualify for inclusion in the calculation of Eq. (8), but it may or may not currently execute it. Thus, the weighting coefficient comprises a set of six variables

$$maa_{i,l} \in \{P_{same} \quad D_{same} \quad F_{same} \quad P_{diff} \quad D_{diff} \quad F_{diff}\} \quad (9)$$

from which only one is chosen to represent the interaction. The designer predefined values of these variables can be used to tune the methodology between flocking and dispersing the swarm according to the application. The first letter "P", "D", or "F" is the basic action of the currently calculated strategy's strength, while the subscript denotes whether the other agent's executed strategy is the same, or different.

The second term of Eq. (2) defines the intruder effect $Sb_i^k(t)$ on strategy i of agent k. Again, certain modifications on Eq. (4) have been applied to conform to the logic of the proposed methodology.

$$Sb_i^k(t) = \begin{cases} 0, & I_k^M(t) = 0 \\ \dfrac{b}{I_k^M(t)} \displaystyle\sum_{m=1}^{I_k^M(t)} mai_{i,m} \varphi^{k,m}, & I_k^M(t) > 0 \end{cases} \quad (10)$$

where $I_k^M(t)$ is the number of all the intruders within the detection radius. The usage of the previous concentration, as in Eq. (4), is no longer valid, since this may be a newly added strategy, thus the agent-to-intruder interaction weighting coefficient $mai_{i,l}$ is introduced in order to model the effect of an enemy appearance against the currently calculated strategy. In this study, it has been determined that four designer-tuned variables may very well describe this interaction, based on the type of the currently calculated strategy (denoted as always by the letters "P", "D", and "F"). Concerning the "D" strategy, there is a distinction on its value based on whether the currently calculated strategy strength is about the same ("D_{same}") or a different ("D_{diff}") intruder, than the one currently interacting with the agent.

$$mai_{i,m} \in \{D_{same} \quad P_{diff} \quad D_{diff} \quad F_{diff}\} \quad (11)$$

Finally, the decay rate follows the form of Eq. (6) and weights the previous strength of the same strategy by the negative decay factor dr, where $0 \leq dr \leq 1$, but now a different decay rate is selected based on the validity of the strategy.

$$Sc_i^k(t) = \begin{cases} -dr_{nv} \cdot S_i^k(t-1), & \text{strategy } i \text{ is invalid} \\ -dr_{cv} \cdot S_i^k(t-1), & \text{strategy } i \text{ is valid} \\ 0, & \text{newly formed strategy} \end{cases} \quad (12)$$

A strategy is deemed valid if, at the current timestep, the target agent or intruder exists in the respective radius. The decay rate coefficient for invalid strategies dr_{nv} and for valid strategies dr_{cv} is selected accordingly, while $0 \leq dr_{cv}, dr_{nv} \leq 1$. If strategy i is newly formed, then the decay rate component is zero.

The final activation is given by Eq. (7) with $\sigma = 0.5$, but a step counter increases by one for every consecutive timestep that the strategy remains in an invalid state. When that counter reaches a predefined strategy validity threshold Sv_{th}, the strategy activation equals zero for every future timestep until its target is detected once again. Until this happens, the strategy is not eligible for selection. The validity threshold may be perceived as a blind follow rule, because, in cases where an engaged intruder is lost (possibly due to climatic difficulties or terrain harshness), the agent will move towards the last known intruder position for a limited time, a fact which may help in engaging the intruder again. When all calculations have been concluded, the strategy activation vector is formed.

$$\mathbf{A}_k(t) = [A_1^k(t) \quad A_2^k(t) \quad \ldots \quad A_{N_k(t)}^k(t)] \quad (13)$$

where $N_k(t)$ is the total number of agent k strategies available at the current timestep. As mentioned earlier, the strategy with the highest activation is selected to be executed at the current time instant, concluding the algorithm.

The standard AIS algorithm is expanded to include the new approach as shown in Algorithm 1, which runs independently on every agent at each discrete time instant:

6. Case studies

In this section, a series of case studies will be discussed, where the proposed methodology will be compared against two decentralized algorithms, and a centralized one. More specifically, the standard AIS methodology described in Section 4 and the modified AIS briefly explained in Section 5 will be employed as the decentralized rivals, while a standard PSO implementation will take up the role of the centralized opponent. In order to use PSO as a centralized approach, the optimization problem should be formulated accordingly, as described in the next sub-section.

6.1. Centralized PSO for swarm formation control

PSO is a well-known optimization algorithm, originally proposed by Kennedy and Eberhart [45]; its effectiveness, simplicity and speed make it ideal for use in applications where computational cost is a critical parameter. PSO has been widely used, while a long list of modifications in the standard algorithm have been published in the literature, since its original development in 1995, in order to suit certain applications. The methodology models the motion dynamics of a swarm of organisms, the main idea originating from flocks of birds and fish schools.

The algorithm encodes a population of particles which represent possible solutions within the problem's hyper-dimensional input space, which are driven towards the optimal solution while exchanging local and global information among them. Each particle updates its position by taking into account its own best position, the global best position and a stochastic parameter, which is able to avert the algorithm from becoming trapped in local minima. An iterative procedure minimizes the cost until a certain preset stop criterion has been reached.

In this work, a local best PSO algorithm was used [46,47]; furthermore, the formulation of the optimization problem has been tailored to meet the need of centralized formation control applications [48–50]. To be more specific, the objective function to be minimized by PSO comprises three conflicting goals and a number of constraints. More specifically, let us assume a swarm of K agents, where some of its members are detecting $M(t)$ intruders at time point t. Then, we may define the matrix containing all Euclidean distances $d_{aa}^{k,l}(t)$between any pair of agents k and l, where $k = 1, 2, \ldots, K$ and $l = 1, 2, \ldots, K$.

$$\mathbf{D}_{aa}(t) = \begin{bmatrix} 0 & d_{aa}^{1,2}(t) & \ldots & d_{aa}^{1,K}(t) \\ d_{aa}^{2,1}(t) & \ldots & \ldots & \ldots \\ \ldots & \ldots & 0 & d_{aa}^{K-1,K}(t) \\ d_{aa}^{K,1}(t) & \ldots & d_{aa}^{K,K-1}(t) & 0 \end{bmatrix} \in \mathbb{R}_{\geq 0} \tag{14}$$

Take note that this is a square and symmetric matrix with a zero diagonal. Given that at time instance t there exists a non-zero number of detected intruders $M(t)$, the set of agents closest to those intruders, by means of the Euclidean distance, is defined as

$$P_{cl}(t) = \left\{ a_1 \quad a_2 \quad \ldots \quad a_m \right\}, 1 \leq m \leq M(t) \tag{15}$$

where a_m is the identification tag of the agent closest to the mth intruder. In this case, the objective function f_{pso} comprises three terms and is formed as follows:

$$f_{pso} = w_1 \frac{1}{g_1(t)} + w_2 g_2(t) + w_3 \frac{1}{g_3(t)} \tag{16}$$

where w_1, w_2 and w_3 are weights used to induce the desirable balance between the three goals. The first term g_1 is the minimum Euclidean distance between all pairs of agents $k = 1, 2, \ldots, K$ and $l = 1, 2, \ldots, K$ comprising matrix $\mathbf{D}_{aa}(t)$, excluding the elements of the main diagonal which may interfere with finding the minimum value, as well as the elements belonging to the agents closest to the detected intruders, so that the engaging agents may not alter their course away from the intruders.

$$g_1(t) = \min \left\{ d_{aa}^{k,l}(t), k, l = 1, 2, \ldots, K \right\}_{\substack{k \neq l \\ k,l \notin P_{cl}(t)}} \tag{17}$$

The second term g_2 sums the Euclidean distances $d_{ai}^{m}(t)$ between the currently detected intruders $M(t)$ and the agents closest to those intruders

$$g_2(t) = \begin{cases} 0, & M(t) = 0 \\ \sum_{m=1}^{M(t)} d_{ai}^{m}(t), & M(t) > 0 \end{cases} \tag{18}$$

The third term $g_3(t)$ calculates the sum of distances between the current and the H previous positions of each agent, excluding the set of agents closest to intruders, divided by the number of remaining agents $K - M(t)$.

$$g_3(t) = \frac{\sum_{a_k \notin P_{cl}(t)} \sum_{h=1}^{H} d_{self}^{a_k}(t-h)}{K - M(t)} \tag{19}$$

where $d_{self}^{a_k}(t-h)$ is the distance between an agent's current position and the position the same agent has been at time instance $t - h$. The effect of the third term is to force any agent that is not currently engaging an intruder, to travel the maximum possible distance away from its own previous positions, which will come to achieve two ends. Firstly, all patrolling agents will explore the area in a more effective way, since they are not allowed to return to a position which they held in the near past and, secondly, they are not allowed to reduce their exploration speed or stop moving, since this term is to be maximized. This latter event has been noticed in some test cases, where the PSO algorithm favored balancing the objective function terms by keeping several agents stuck on the map edges, while the others were engaging intruders or patrolled, a fact which rendered some agents useless.

The optimization procedure tries to determine the velocities $v_k(t)$ and rotational angles $ra_k(t)$ of all agents K that will maximize $g_1(t)$ and $g_3(t)$, so that the distribution of patrolling agents allows for the optimal coverage of the search space, while also trying to minimize $g_2(t)$, in order for the engaging agents to closely follow the currently detected intruders, if any.

The constraints include the inability of any agent to move outside the map boundaries, while also maintaining speed and rotational angle restrictions to conform to the hardware capabilities and specifications.

6.2. Environment and tuning parameters

In this work, the max speed of all UAVs is restricted to 50 km/h, unless otherwise noted, while turning angles are restricted to ±20°. The sampling time for all following case studies is set to 1s for calculation simplicity. The rivaling algorithms are the static strategy AIS (SS AIS), the thymus-enhanced static strategy AIS (TE-SS AIS), the proposed enhanced decay variable strategy AIS (ED-VS AIS) methodology, and the centralized PSO algorithm. The main parameters for each methodology are shown

Algorithm 1

```
1:  Initialization
        New pool of strategies along with their strengths, activations and concentration values is
        copied as is from the data of the previous time instant. At $t=0$ the only strategy in the pool
        is Patrol with an activation of 0 and a concentration of 1.
2:  Sweep environment to get positions of agents within communication range and intruders within
    detection range
3:  Form an individual D-type strategy for every intruder detected and an F-type strategy for every
    agent detected. Crosscheck with the existing strategy pool and append any newly found strategies.
4:  For each strategy in the pool:
        BEGIN loop
4a:         Communicate with nearby detected agents and obtain their strategy concentrations
            (only for the strategy being calculated)
4b:         Calculate new strategy strength through Eqs. (2),(8),(10), and (12)
4c:         Calculate new activation through Eq. (7)
        END loop
5:  Form the strategy activation vector of the agent through Eq. (13)
6:  Execute the strategy with the highest activation score.
```

Table 2
Tuning parameters

Description	Symbol	Value
Common parameters for all AIS-based methodologies		
Strength term *Sa* coefficient	a	0.5
Strength term *Sb* coefficient	b	1.0
Common parameters for SS and TE-SS AIS		
Decay rate coefficient	μ	0.9
Mutual interaction weighting coefficient	$m_{i,j}$	see Table 1
TE-SS AIS Parameters		
Critic threshold	C_{th}	2
ED-VS AIS Parameters		
Decay rate coefficient (valid strategies)	dr_{cv}	0.9
Decay rate coefficient (invalid strategies)	dr_{nv}	0.99
Strategy validity threshold	Sv_{th}	5
Agent-to-agent interaction weighting coefficient	$maa_{i,l}$	see Table 3
Agent-to-intruder interaction weighting coefficient	$mai_{i,m}$	see Table 3
PSO Parameters		
Number of previous positions taken into account for $g_3(t)$ calculation	H	10
Weighting coefficient of $g_1(t)$	w_1	4
Weighting coefficient of $g_2(t)$	w_2	1
Weighting coefficient of $g_3(t)$	w_3	50

in Table 2. The mutual interaction weighting coefficient values, which are common for the cases of SS and TE-SS AIS, are given in Table 1. As it can be seen, a different coefficient value applies based on the executed strategy of agents i and j. A special case exists that needs to be noted as per the original publication. The coefficient has a different value if both agents execute strategy “F", depending upon whether they follow the same ($p = q$) or a different ($p \neq q$) agent. Table 3 depicts the agent-to-agent and agent-to-intruder interaction weighting coefficient values regarding the proposed ED-VS AIS methodology.

All methodologies were tuned by trial and error with a set of purposes in mind. At first, at least one agent should follow each intruder for the whole simulation time after detection, if constraints allow it. In cases where this is not possible (i.e. intruder is faster than the agent), the agent may continue moving towards the last known direction of the intruder, until a predefined criterion has been met, before reverting to patrol mode. More specifically, the agent is allowed to keep the same direction for a predefined period of time (in our study we use a 5 timesteps

Table 3
Mutual interaction coefficients for ED-VS AIS methodology

Symbol	Value
Agent-to-agent interaction weighting coefficient $maa_{i,l}$	
P_{same}	0.0
D_{same}	−5.5
F_{same}	−0.1
P_{diff}	+0.4
D_{diff}	+0.6
F_{diff}	+0.2
Agent-to-intruder interaction weighting coefficient $mai_{i,m}$	
D_{same}	+0.1
P_{diff}	−0.1
D_{diff}	−0.1
F_{diff}	−0.1

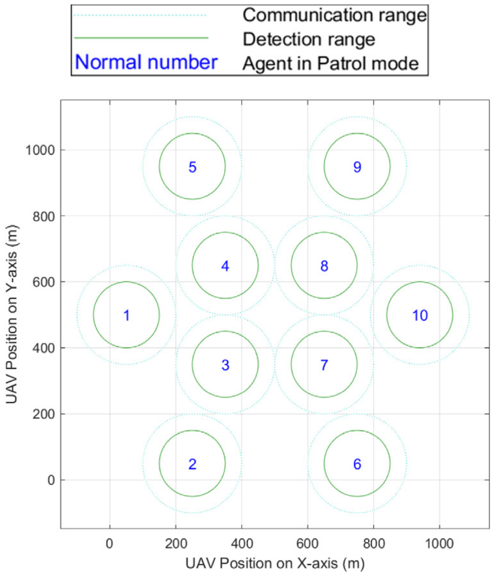


Fig. 3. Initial positions of 10 agents on a grid.

threshold) or until it has reached the last known intruder position, whichever comes first. The search space should have the maximal coverage without affecting the agent's ability to engage intruders.

In every test case presented, the UAVs are initialized in a grid formation covering the whole area as depicted in Fig. 3 and are let to roam randomly for 30 s before any enemy appears on the map. The same random movements are used in all algorithms per case, thus forcing the initial positions of the agents to be the same in the four runs of each test case. As far as the intruders are concerned, the full route they follow throughout the simulation is exactly the same for all algorithms, in order to make the performance metric scores of all algorithms tested on a specific case directly comparable. The supplementary material contains a video for each test case, showing a whole simulation, so that the reader may observe the performance of each algorithm in detail. The zero timestep shown in the provided videos starts after the 30s roaming.

6.3. Performance metrics and statistical significance testing

To evaluate each case study, five performance metrics are used. More specifically, there is one performance metric with respect to the exploration goal (denoted G1), as expressed by the sum of $g_1(t)$ values across the whole simulation. Another performance metric (denoted G2) expresses the goal of detecting and engaging intruders, that being the sum of $g_2(t)$ values. The sum of all $g_3(t)$ values, signifies a mobility criterion (denoted G3), to show whether the algorithms correctly roam the free space instead of keeping a number of available agents standing still. The intruder detection count (IDC) shows the number of intruders that have been detected by the agents at the end of the simulation. The last performance metric is the mean step duration (MSD), which, as the name implies, is the average of each algorithm's calculation durations $cd(t)$ over the total duration T of the whole simulation. Each simulation has a duration of $T = 120$ s corresponding to 120 timesteps.

Since all of the competing methodologies contain non-deterministic elements, each simulation has been performed 200 times, while each execution was provided with a different random set of initial values. In this manner, a set of 200 values has been derived for each performance metric per methodology and per case. Based on the results, a t-test between the proposed methodology and each one of its rival methodologies was implemented, in order to determine whether there is a statistically significant difference between the performance metric mean values. The null hypothesis is that the results produced by the two competing methodologies are generated by populations with the same mean.

It should be noted that the desktop computer on which all simulations and respective measurements took place is equipped with an Intel Core i5-4690K CPU and 12 GB DDR3 RAM.

6.4. Test cases descriptions and results

Six test cases have been designed to cover different major scenarios that a surveillance UAV swarm could come up against. This section presents brief descriptions for each test case, along with the corresponding results.

- Test case 1: Single intruder
  A single intruder appears at a random position and freely roams the map, while a swarm of 10 agents tries to engage it. The intruder detection radius is 100 m, the agent communication radius is 150 m and the speeds of all UAVs are limited to 50 km/h.
- Test case 2: Multiple intruders
  Five intruders have been deployed at random positions in order to evaluate the behavior of each algorithm in a multiple intruder scenario, while keeping the same number of agents as before. The intruder detection radius is 100 m, the agent communication radius is 150 m and the speeds of all UAVs are limited to 50 km/h.
- Test Case 3: Ranges modification — Wider detection area
  This test case employs 10 agents against 5 intruders, but now the intruder detection radius has been increased five times from 100 m to 500 m. The agent communications radius and the speed limits remain the same at 150 m and 50 km/h, respectively.
- Test Case 4: Ranges modification — Wider communication area
  In this test case the agent communication radius is increased from the original 150 m to the much wider 500 m. The rest of the parameters remain as they were given for test case 2. More specifically, the swarm employs 10 agents which are put against 5 intruders, the intruder detection radius is 100 m, and the speed is limited to 50 km/h.
- Test Case 5: More intruders than agents
  In this case we have reversed the populations between the agents and the intruders forming a swarm of five agents against ten intruders. The rest of the parameters are 100 m for the intruder detection radius, 150 m for the agent communication radius, and a 50 km/h speed limit for all UAVs.

- Test Case 6: Intruders faster than agents
  This last case has been designed so that the intruders outperform the agents in terms of speed. All intruder motion is now limited at 55 km/h, while the agents are hindered with a speed limit of 45 km/h. The swarm consists of ten agents and five intruders. The intruder detection radius is 100 m, and the agent communication radius is 150 m.

For each case 1-6:

- The *p*-values, means, and standard deviations for all performance metrics are given in Tables 4–9, respectively
- A randomly selected simulation from the 200 runs is shown in Videos 1–6, respectively
- The final positions of all UAVs at the end of the same randomly selected simulation are depicted in Figs. 4–9, respectively

6.5. Discussion

The scores in Tables 4–9 illustrate that the proposed methodology is producing high quality results, overpowering its rivals in actual performance in the majority of the performance metrics. Looking at the obtained G1 metric which accounts for the exploration capabilities of the swarm when in patrol mode, it can be deduced that in every case examined the ED-VS AIS methodology explores the area in a more efficient way than SS AIS and TE-SS AIS, while in case 5 it seems that it performs even better than its centralized rival. The *p*-values show that this conclusion is statistically significant with a confidence level of 96% or higher, in all cases. The result of case 5 can be explained by the fact that PSO is tuned to perform in an optimal way for the surveillance scenario. Thus, it will prefer to direct the swarm on the targets instead of exploring the area. One point to consider here is that the PSO algorithm may be tuned to favor exploration, but at the cost of providing far inferior results regarding the other metrics. A middle ground is very difficult to be obtained in terms of PSO tuning, so PSO cannot perform desirably in all cases. On the other hand, the immune system is inherently designed for better exploration and the proposed approach further improves this capability, as one can verify through the provided statistical analysis.

Scores of G2 metric, which calculates how well the swarm of agents detects and follows the intruders, show that in most cases ED-VS AIS performs comparably well, or outperforms the other tested schemes with a high statistical significance score. The figures and videos, clearly illustrate encounter management has improved substantially by the proposed method compared to its rival AIS-based methodologies. The most important fact, which is evident when coupling the results of the G2 metric scores with the IDC metric scores, is that the proposed scheme always engages significantly more threats for more time compared to any one of the other two AIS-based schemes, a fact which has the utmost importance in a surveillance or a search & rescue mission. Compared to PSO, the proposed methodology obtains better scores in the G2 metric in all but one case. More specifically, in case 3, in which the detection area is widened to the point where all agents detect all intruders, it is only logical that a centralized approach already having the benefit of total command in terms of communication, may position the agents in a better way than a methodology which has a specific communication radius between swarm members and, at the time, each UAV has to decide its own route, regardless of the choices of distant agents.

The mobility criterion measured by the G3 metric, shows that ED-VS AIS is capable of keeping the agents on the move without going back to recently visited parts of the control area. It is not surprising that mobility scores are somewhat worse compared to the other two AIS-based approaches, as the difference in IDC scores becomes larger. This can be explained by the fact that any agent pursuing an intruder is mostly led by the intruder's trajectory; thus, if an intruder decides to turn back to a previously visited position the agent will follow and this will produce a worse G3 metric score. As noted earlier, however, the most important mission of the swarm is to track intruders. The G3 metric is very important when not engaging an intruder and the fact that the proposed methodology is comparable to the AIS-based rivals and, at the same time, is far better than PSO in most cases, shows that the algorithm takes into account the significance of patrolling different areas.

Finally, it should be noted that the superior performance of the proposed approach over the remaining AIS-based schemes comes at a low cost as far as computational times are concerned; compared to the different AIS-based schemes, the overhead in CPU cost is low as shown by the MSD metric, thus allowing for real-time implementation without requiring the UAVs to be equipped with expensive high performance on-board processors. On the other hand, ED-VS AIS is 8 to 20 times faster compared to the centralized PSO scheme, while its performance is comparable or in some cases, even superior.

An in-depth per case analysis can be found in the included appendix, where more details are provided regarding the behavior of the compared methodologies for each case.

7. Conclusions

In this work a novel decentralized methodology for strategy assignment to swarms of autonomous vehicles has been presented. The proposed ED-VS AIS methodology is based on the artificial immune system theory which uses local exchange of information between neighboring swarm members. Through this process, a suboptimal solution is calculated, pertaining to the problem of selecting the proper strategy so that the whole swarm performs in a desirable manner.

The presented algorithm is evaluated in several test cases designed around an area surveillance scenario, where the goal is to optimally explore the area, while also prioritizing how to engage any enemies found. The performance of the proposed methodology is compared against three rivaling algorithms: two decentralized algorithms based on the AIS theory (SS AIS and TE-SS AIS), and one centralized method based on the well-known PSO algorithm. *t*-test statistical analysis has been performed in order to certify the statistical significance of the results, which have shown that the proposed methodology has a number of advantages over its competitors:

- It avoids flocking by assigning the least amount of swarm members to each enemy found.
- It has an increased probability of producing a high exploration efficiency, by maximizing the availability of free roaming members.
- Members that are not engaging enemies exploit the option of following a friendly swarm member in order to assist in cases where the enemies have the advantage.
- Mobility and distance of the swarm members are always kept at a high level, accommodating a better exploration effect.
- The ability to follow the trace of an enemy who has managed to escape provides an added value to the algorithm, increasing its effectiveness in surveillance scenarios.

There are also some cases where the proposed methodology obtains a better score than its centralized rival, despite the fact that the former formulates and solves an optimization problem based on the positions of all agents at each time instant. One of the most

Table 4
Performance metrics for Test Case 1

	G1↗(km)			G2↘(km)			G3↗(km)			IDC↗(#)			MSD↘(ms)		
	p-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std
SS AIS	0.00	7.11	2.99	0.37	2.73	1.25	0.00	719.63	24.62	0.00	0.73	0.45	0.00	9.05	1.25
TE-SS AIS	0.00	8.75	2.13	0.00	3.45	1.38	0.00	719.87	22.03	0.00	0.84	0.37	0.20	9.70	0.96
ED-VS AIS		10.08	1.73		2.82	0.96		711.65	17.85		0.99	0.12		10.00	3.51
PSO	0.00	36.62	2.03	0.00	3.10	1.08	0.00	603.04	17.95	0.66	0.99	0.10	0.00	179.71	18.57

Arrow direction indicates whether metric should be maximized or minimized.

Table 5
Performance metrics for Test Case 2

	G1↗(km)			G2↘(km)			G3↗(km)			IDC↗(#)			MSD↘(ms)		
	p-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std
SS AIS	0.00	8.26	4.92	0.00	13.90	3.32	0.00	567.02	45.11	0.00	2.81	0.86	0.00	10.55	1.06
TE-SS AIS	0.00	10.30	3.41	0.02	14.19	3.58	0.00	572.07	42.67	0.00	2.95	0.76	0.00	10.72	1.06
ED-VS AIS		13.64	4.25		15.07	4.11		525.66	55.82		4.43	0.66		15.96	2.94
PSO	0.00	41.68	3.77	0.00	18.61	3.32	0.00	410.01	36.97	0.00	4.84	0.42	0.00	294.91	29.40

Arrow direction indicates whether metric should be maximized or minimized.

Table 6
Performance metrics for Test Case 3

	G1↗(km)			G2↘(km)			G3↗(km)			IDC↗(#)			MSD↘(ms)		
	p-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std
SS AIS	0.00	4.52	1.05	0.00	83.83	23.84	0.00	429.02	27.02	0.00	4.59	0.64	0.00	11.98	1.37
TE-SS AIS	0.04	6.00	2.91	0.00	83.08	23.86	0.00	408.75	23.25	0.00	4.67	0.74	0.00	12.30	1.56
ED-VS AIS		6.72	2.79		49.46	16.53		403.83	9.08		4.95	0.23		40.30	4.30
PSO	0.00	45.00	3.78	0.00	32.11	7.95	0.00	341.73	10.53	0.00	5.00	0.00	0.00	325.45	22.29

Arrow direction indicates whether metric should be maximized or minimized.

Table 7
Performance metrics for Test Case 4

	G1↗(km)			G2↘(km)			G3↗(km)			IDC↗(#)			MSD↘(ms)		
	p-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std
SS AIS	0.00	8.22	4.36	0.00	12.43	2.97	0.00	541.42	48.51	0.00	3.60	0.72	0.00	14.83	1.14
TE-SS AIS	0.00	6.80	3.63	0.05	15.84	4.35	0.17	517.73	69.92	0.00	3.07	0.87	0.00	14.98	1.98
ED-VS AIS		13.66	3.63		14.96	4.43		526.43	56.44		4.05	0.77		29.79	6.60
PSO	0.00	41.68	3.77	0.00	18.61	3.32	0.00	410.01	36.97	0.00	4.84	0.42	0.00	294.87	29.53

Arrow direction indicates whether metric should be maximized or minimized.

Table 8
Performance metrics for Test Case 5

	G1↗(km)			G2↘(km)			G3↗(km)			IDC↗(#)			MSD↘(ms)		
	p-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std
SS AIS	0.00	64.35	24.72	0.00	22.32	6.99	0.00	196.66	40.33	0.00	2.43	1.15	0.00	4.77	0.58
TE-SS AIS	0.01	68.84	25.22	0.00	41.34	17.80	0.00	164.42	54.45	0.00	3.47	1.79	0.00	4.90	0.73
ED-VS AIS		77.32	38.11		28.94	9.84		87.11	44.57		5.36	0.97		10.14	1.57
PSO	0.00	56.04	33.17	0.00	36.61	12.87	0.00	62.88	37.72	0.00	5.62	0.80	0.00	186.56	16.75

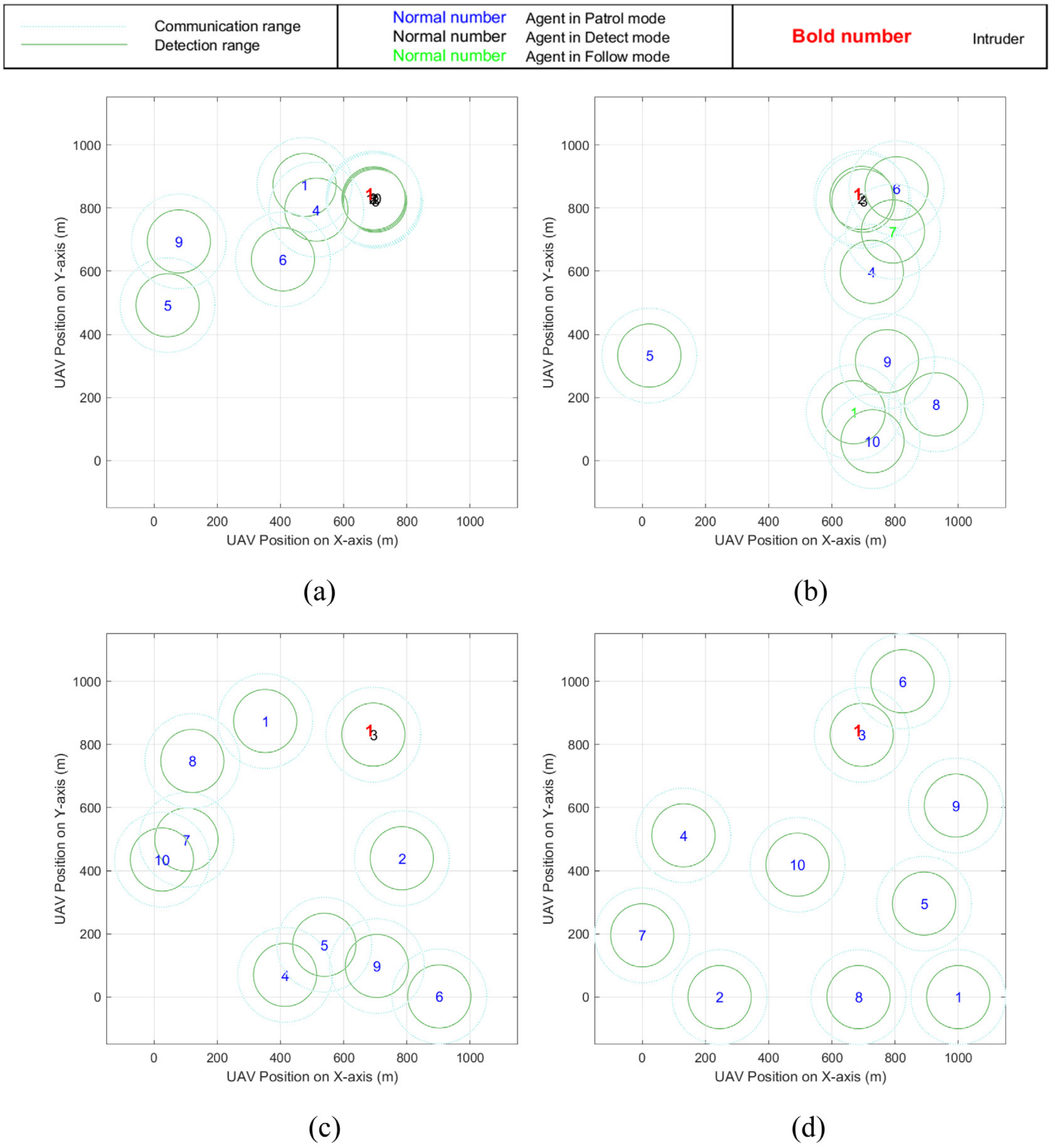
Arrow direction indicates whether metric should be maximized or minimized.

Table 9
Performance metrics for Test Case 6

	G1↗(km)			G2↘(km)			G3↗(km)			IDC↗(#)			MSD↘(ms)		
	p-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std	*p*-value	Mean	Std
SS AIS	0.00	8.85	3.20	0.19	17.40	4.97	0.00	548.77	35.74	0.00	2.08	0.64	0.00	10.11	0.94
TE-SS AIS	0.00	9.40	2.65	0.00	16.89	3.43	0.02	543.76	30.29	0.01	2.11	0.62	0.00	10.36	0.93
ED-VS AIS		10.93	3.06		17.96	3.69		536.74	33.50		2.37	0.63		17.55	2.83
PSO	0.00	34.36	2.79	0.00	22.72	3.72	0.00	431.60	35.41	0.16	2.51	0.81	0.00	224.50	21.51

Arrow direction indicates whether metric should be maximized or minimized.

notable points is the fact that ED-VS AIS adds an insignificant computational time overhead to the AIS-based rivals, but still keeps the MSD to 1/10th when compared to the centralized PSO approach.



(a)

(b)

(c)

(d)

Fig. 4. Final swarm positions of (a) SS AIS, (b) TE-SS AIS, (c) ED-VS AIS, and (d) PSO algorithms for Test Case 1.

Future research will be directed to the path of applying the proposed methodology to different control problems which could benefit from a robust decentralized approach, like the case of cooperative cargo transport, the control of smart grids, search and rescue scenarios employing UAV swarms, and ad hoc telecommunication networks.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2020.106135.

CRediT authorship contribution statement

Marios Stogiannos: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Alex Alexandridis:** Conceptualization, Formal analysis, Methodology, Project administration, Resources, Supervision, Validation, Writing - original draft, Writing - review & editing. **Haralambos Sarimveis:** Conceptualization, Methodology, Project administration, Resources, Supervision, Writing - original draft, Writing - review & editing.

Acknowledgments

The research work was supported by the Hellenic Foundation for Research and Innovation (HFRI), Greece and the General Secretariat for Research and Technology (GSRT), Greece, under the HFRI PhD Fellowship grant (GA. no. 2731).

Appendix A

The selected test cases have been designed in order to display the advantages and disadvantages of the competing methodologies in a meaningful way. In this appendix a detailed per case analysis of the experimental section can be found.

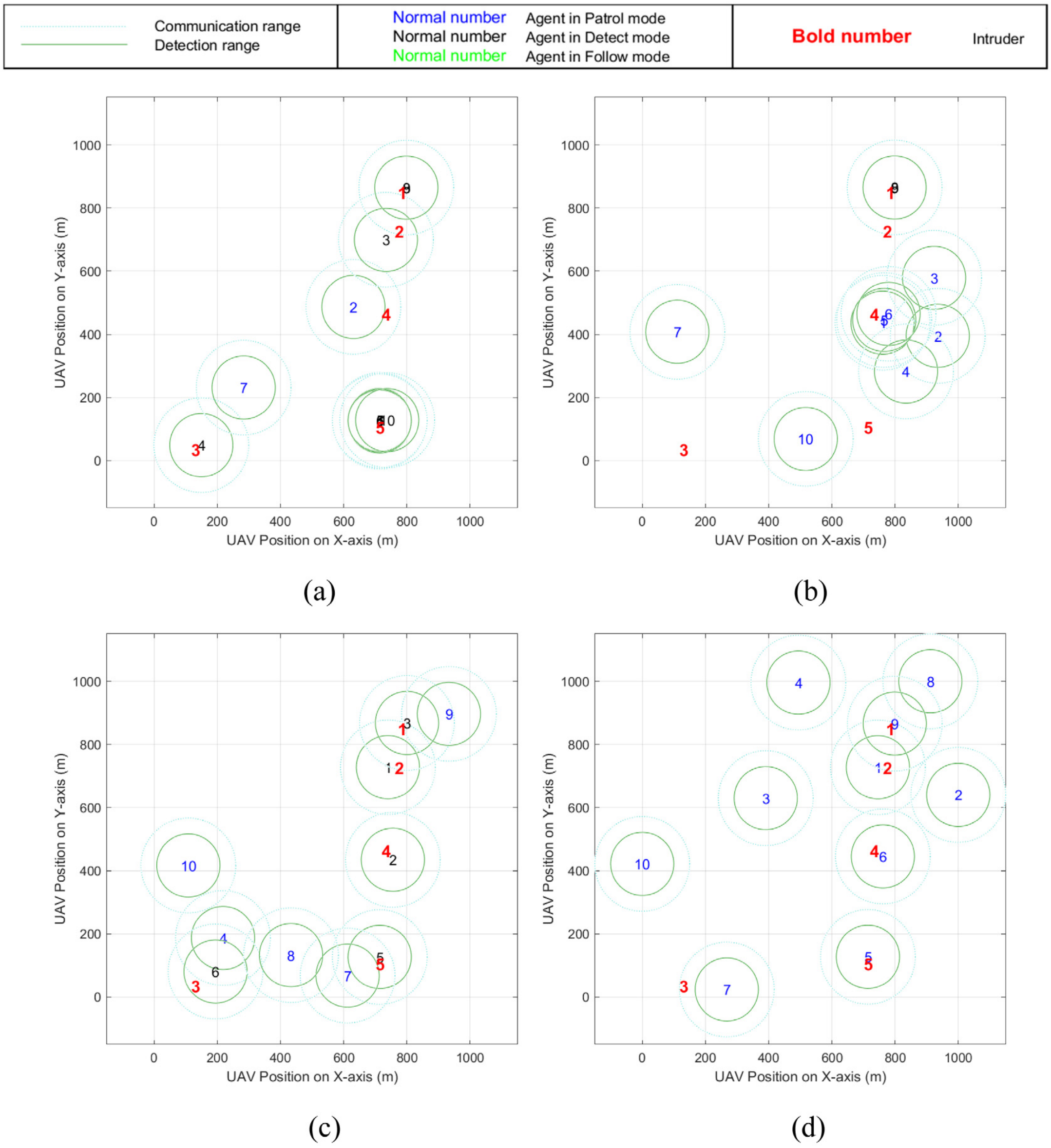


Fig. 5. Final swarm positions of (a) SS AIS, (b) TE-SS AIS, (c) ED-VS AIS, and (d) PSO algorithms for Test Case 2.

A.1. Test case 1: Single intruder

The purpose of such a case is to show how each methodology fares against flocking around a single intruder, when the number of agents detecting the same intruder is more than one. The proposed methodology achieves better exploration performance compared to the other AIS-based algorithms, and is comparable with respect to the mobility metric. The flocking effect is also kept to a minimum; in fact, ED-VS AIS is the only decentralized methodology that assigns only one agent to the found intruder. As seen in Fig. 4a and b, the SS AIS assigns five agents, while the TE-SS AIS assigns two. In terms of calculation times it can be seen that the proposed approach adds a very small overhead compared to the other AIS-based algorithms, due to the slight increase in complexity, but it is not significant enough to hinder the real-time strategy assignment process. In fact, the ED-VS AIS method is many times faster compared to the PSO-based centralized approach, while also getting better scores in G2 and G3 metrics with high statistical significance.

A.2. Test case 2: Multiple intruders

In this test case five intruders have been deployed at random positions in order to evaluate the behavior of each algorithm in a multiple intruder scenario. The SS AIS algorithm has a high flocking behavior leaving a very small number of agents to continue patrolling. It can also be seen that not all intruders have been engaged, a fact which can be noticed for the TE-SS AIS methodology as well. The latter fares better on the minimization of flocking compared to the SS AIS. On the other hand, the proposed methodology engages all intruders, while having the best behavior compared to the rest of the AIS-based algorithms, in terms of less flocking and more exploration. Looking at the MSD score of the decentralized algorithms, one can notice that the overhead increase is about the same in absolute values compared to the first test case, which means that it is actually very loosely related to the increase in the number of agents. Compared to the PSO algorithm, the proposed methodology is better in all performance metrics, except the G1 metric denoting the exploration, which is expected, as PSO can form its decision based on significantly more crucial information gathered from the whole swarm. On the

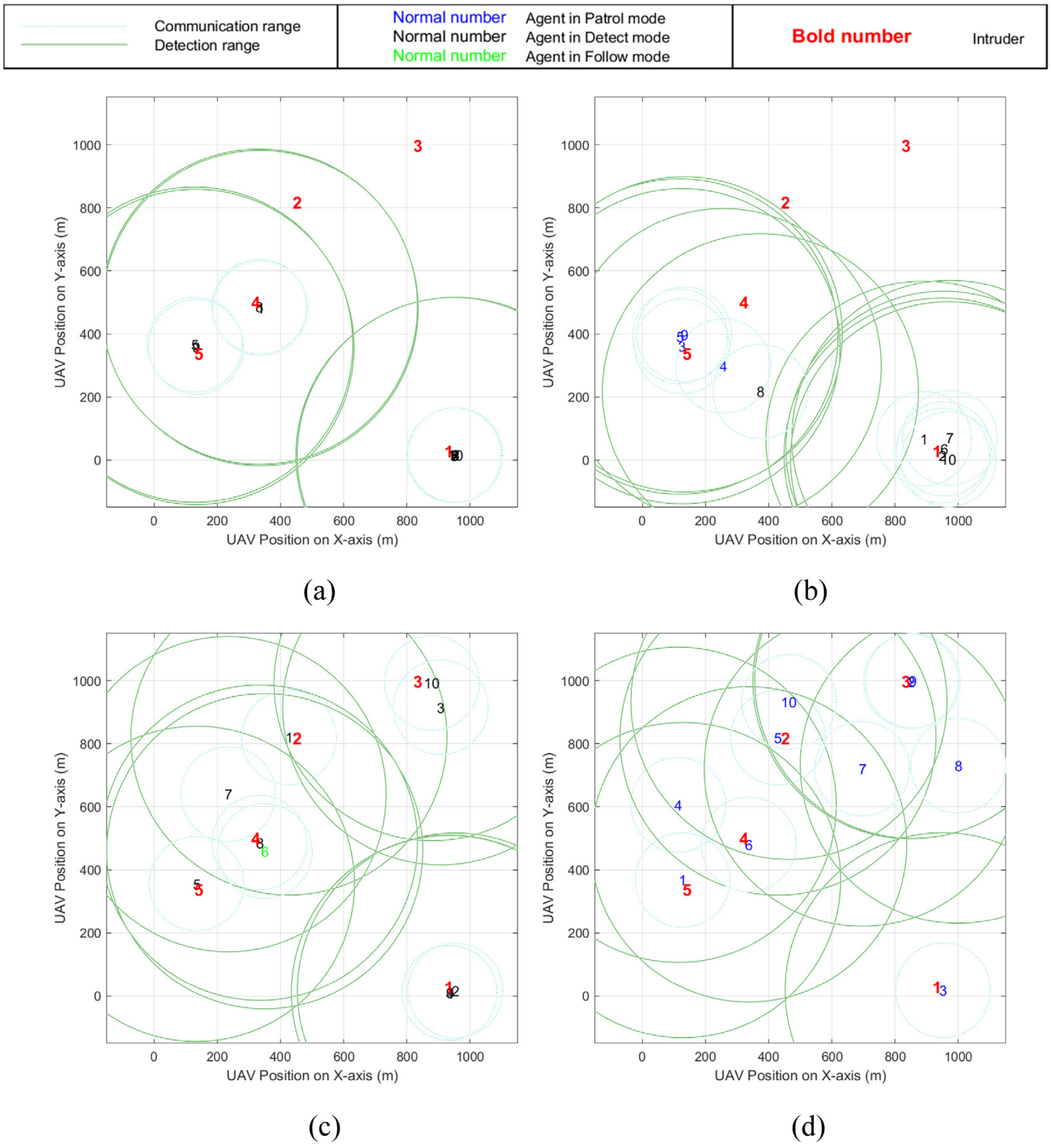


Fig. 6. Final swarm positions of (a) SS AIS, (b) TE-SS AIS, (c) ED-VS AIS, and (d) PSO algorithms for Test Case 3.

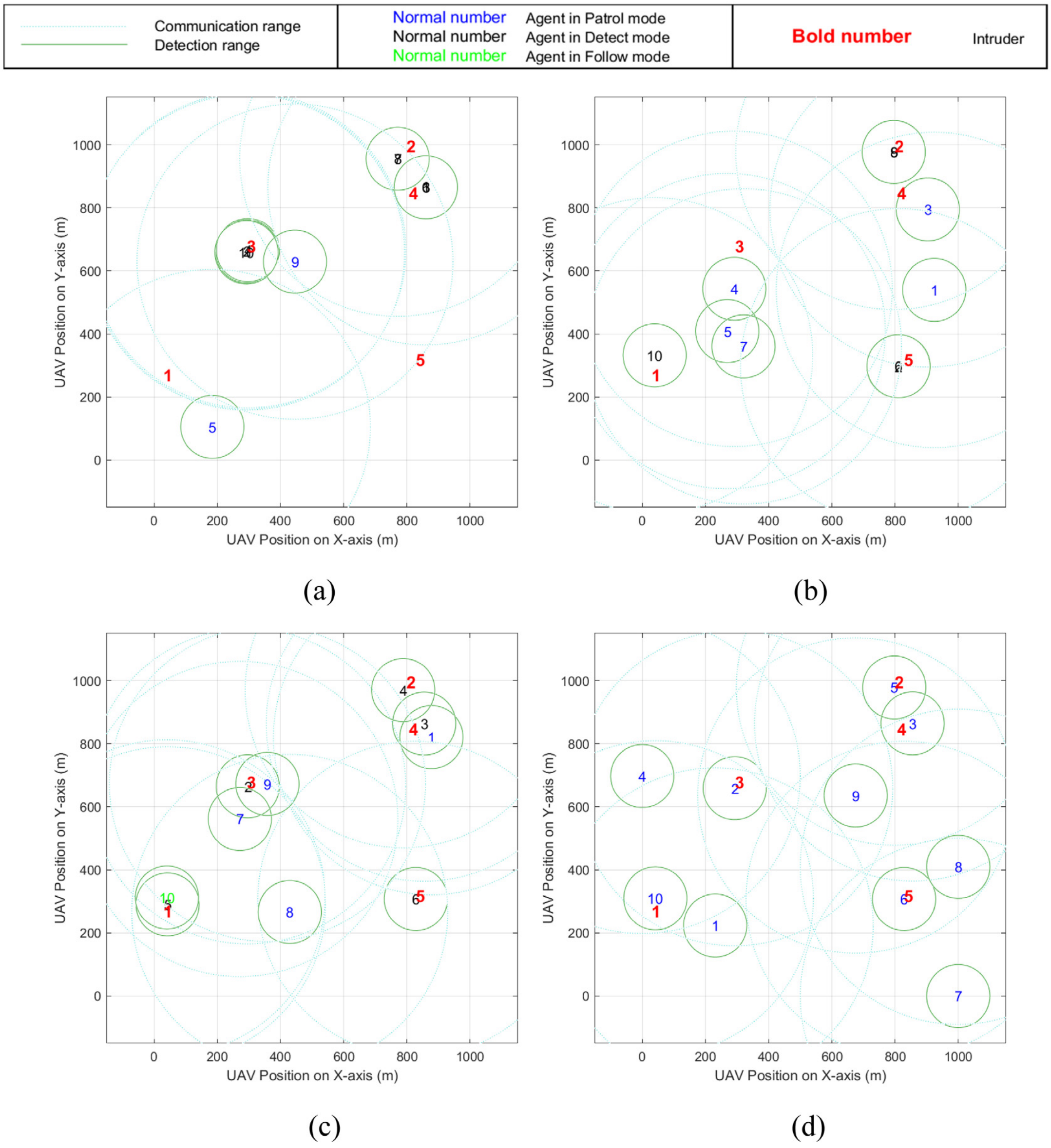
other hand, this comes at a high cost in the MSD metric, which in this case is ten times higher compared to the proposed approach. Comparing the IDC metric scores, one can see the advantageous score obtained by the proposed methodology with an over 50% increase compared to the other AIS-based approaches, a number which brings it very close to the PSO approach.

A.3. Test case 3: Ranges modification — Wider detection area

In this test the algorithms are evaluated on their ability to perform as required and avoid flocking, while being hindered by the fact that all (or most) agents detect multiple intruders. To accomplish that, we have increased the detection radius from 100 m to 500 m. As seen in the respective figures, the SS AIS and the TE-SS AIS algorithms still miss some of the threats, while flocking around a small number of intruders. More specifically, in the SS AIS methodology intruder 3 is totally missed and even intruder 2, which is within detection range, is ignored. The TE-SS AIS algorithm cannot cope at all with the increase in the detection range behaving erratically and flocking around intruder 1, while simultaneously missing detection of intruders 2 and 3. In a way, of course, this behavior is better than the standard AIS methodology, because although it does not detect a number of intruders, it still leaves agents in patrol mode. The proposed methodology detects all intruders, while also keeping flocking to a single intruder at a minimum. In Table 6 it can be seen that the ED-VS AIS algorithm scores better than the other AIS-based methodologies in G1 and G2 metrics, while still being comparable to them in the G3 metric; it is important that this is accomplished with only a small time overhead. The ED-VS AIS algorithm also exploits the F strategy in this case, something which can be regarded as an urge, not to engage the same intruders, but be on an alert state, while still giving some priority to patrolling the area.

A.4. Test case 4: Ranges modification — Wider communication area

This test case employs the opposite logic of the previous one. Now, the communication ranges have been widened to 500 m from their original 150 m radius, which will force the



(a)

(b)

(c)

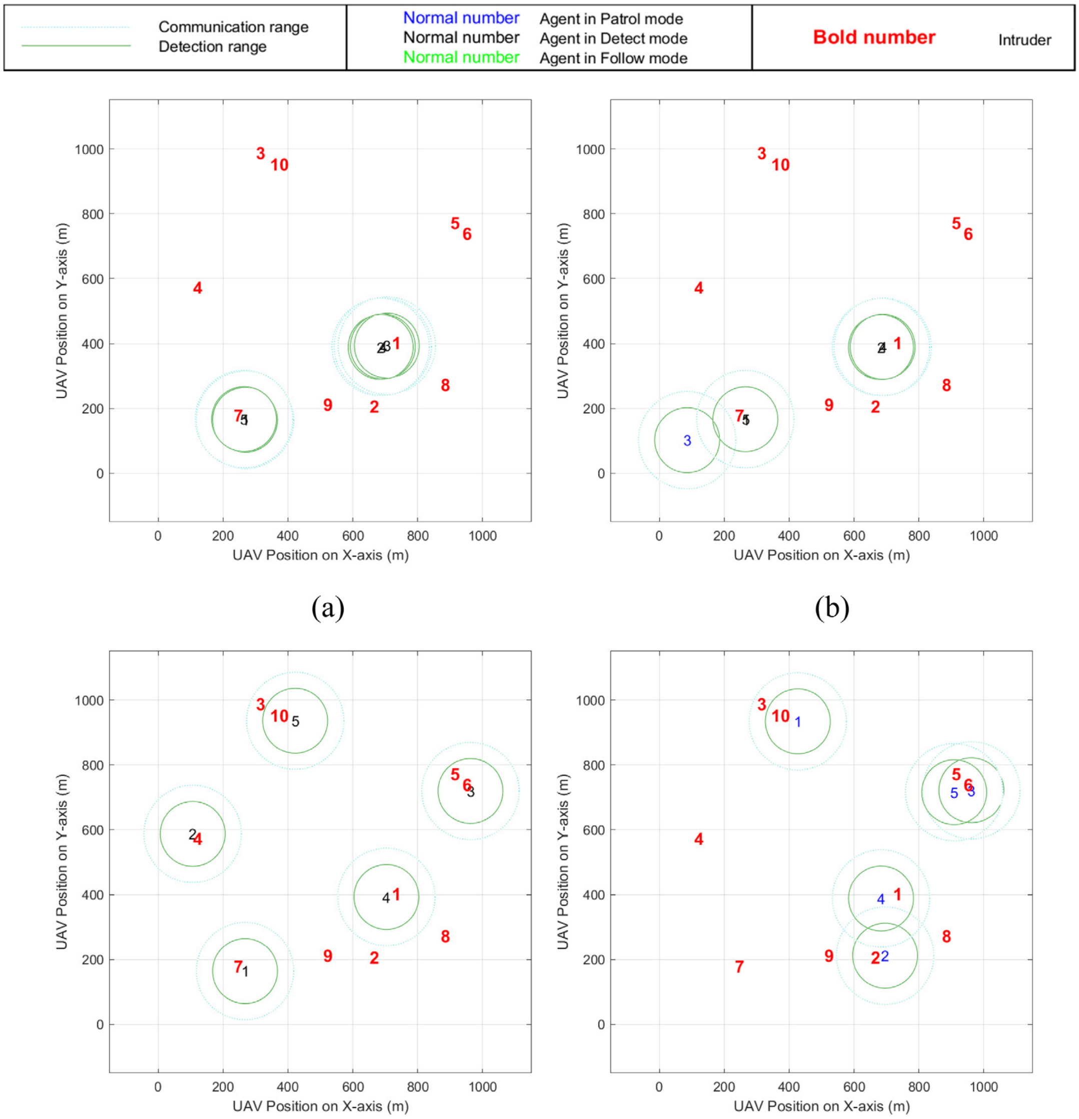
(d)

Fig. 7. Final swarm positions of (a) SS AIS, (b) TE-SS AIS, (c) ED-VS AIS, and (d) PSO algorithms for Test Case 4.

algorithms to an explosion of information between the agents. The purpose is to evaluate the performance of the algorithms on a scenario where most information obtained from the whole area under surveillance is shared among all members of the swarm, coming closer to the logic of centralization. In Fig. 7a, it is obvious that the SS AIS algorithm, despite the abundance of information, misses two intruders, while still not being able to avoid flocking. Fig. 7b shows that one intruder has been missed by the TE-SS AIS methodology too, although it performs better in exploration, but worse in mobility, compared to standard SS AIS. The proposed methodology is able to tackle the explosion of information engaging all intruders, while also leaving four out of five available agents free to patrol. One agent (agent 2) chooses the "F" strategy, an ability seldom exploited by the other AIS-based methodologies. The proposed methodology achieves a better score in G1 metric and similar results in G2 and G3 metrics compared to the other decentralized algorithms. The fact remains that it is the only decentralized scheme, which still engages all intruders, despite the problems stemming from the explosion of information.

A.5. Test case 5: More intruders than agents

In this case we have reversed the populations between the agents and the intruders (five agents against ten intruders) in order to examine how the algorithms cope with a scenario of being overrun with enemies. Fig. 8a shows that the SS AIS algorithm is unable to perform well. Only two intruders are engaged by flocking all available agents around them, leaving the largest part of the enemies unchallenged. The TE-SS AIS methodology fares similarly bad, except that it is able to keep one agent in patrol mode, but the remaining agent is far from any possibility of detecting an intruder, as it is patrolling a remote empty area. The proposed methodology seems to take the lead in this test case, by assigning all the agents to detected intruders, while also keeping the same agents at close distance to more than one intruder. Four out of ten intruders are not engaged, while one more is close to being detected (intruder 3), something quite remarkable for a decentralized algorithm, which only processes local information. The same results are produced by the PSO algorithm, which is expected in this case, since a centralized methodology exploits all available area information, to deploy its agents in



(a) (b)

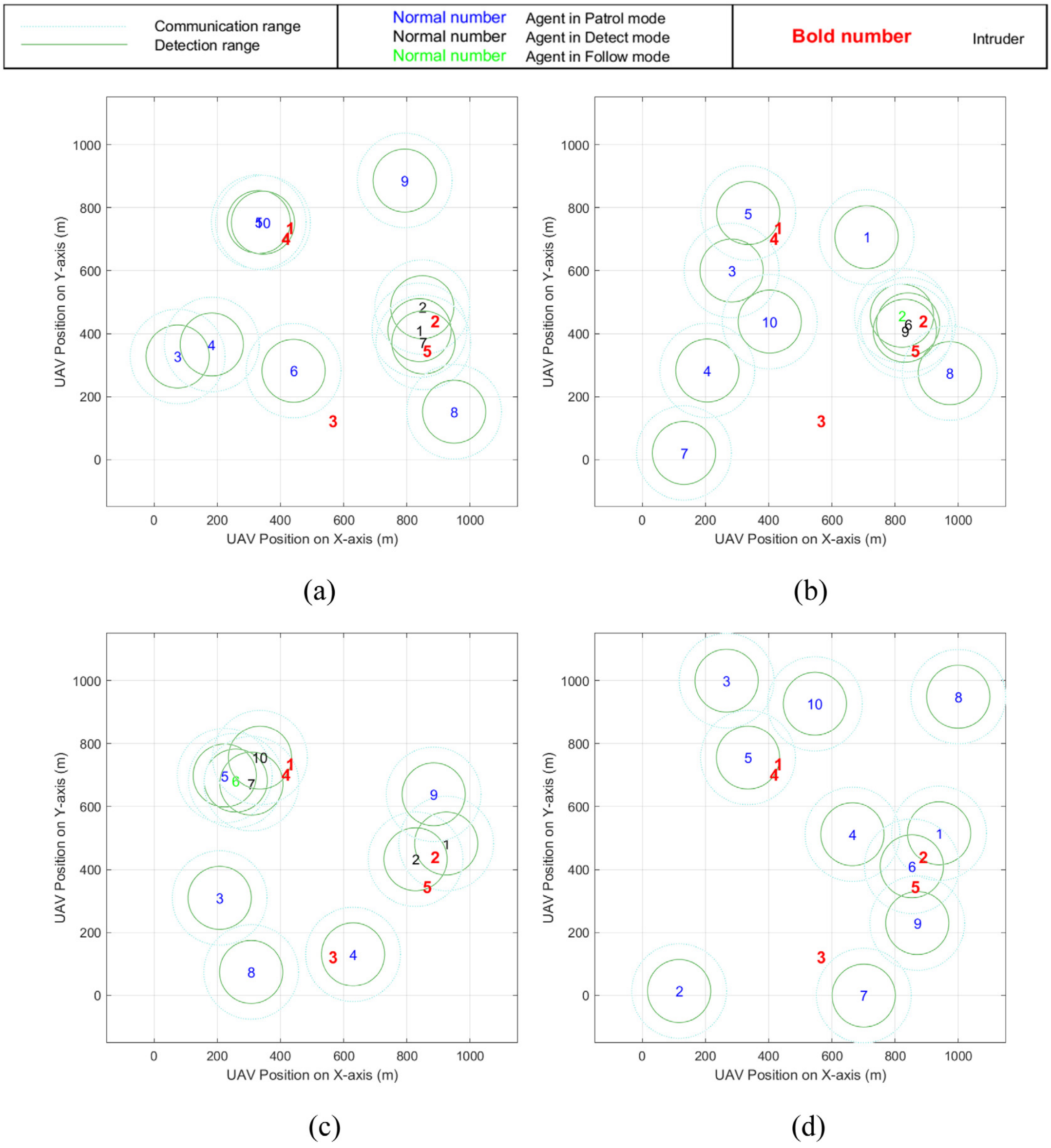
(c) (d)

Fig. 8. Final swarm positions of (a) SS AIS, (b) TE-SS AIS, (c) ED-VS AIS, and (d) PSO algorithms for Test Case 5.

the best formation possible. The performance metrics in Table 8, show comparable scores between the SS AIS and the TE-SS AIS methodologies, but their actual performance is not acceptable as previously explained. The proposed scheme outperforms the centralized PSO algorithm in every metric, except the IDC, which is very close for the two approaches. ED-VS AIS is also better in spreading the agents, so that they are still able to have a wide overview of the area under control. Note that, once again, these results come at a very small fraction of the time needed by the centralized PSO algorithm.

A.6. Test case 6: Intruders faster than agents

This last case is designed to study the response of rivaling algorithms under a scenario where the intruding enemies are faster than the available agents. In this case the agents' speed has been lowered to 45 km/h, while the intruders' speed has been increased to 55 km/h. The scores show that the proposed methodology obtains similar scores to all metrics, while engaging more intruders. Both the SS AIS and the TE-SS AIS miss an intruder and do not actually differentiate their routes or final positions much, a fact showing that TE-SS AIS is not able to perform better than standard SS AIS in such a case. This is also depicted in their score similarity. This time PSO wins all algorithms in the G1 metric providing a better exploration of the area, but not in mobility, something that leaves one intruder roaming free. Fig. 9c shows that the proposed methodology is the only one that is able to keep all intruders under surveillance at the very least, despite the fact that the agents are not fast enough to keep up with them. The TE-SS AIS methodology shows some promising results in this case, but the covered area is comparable to the results of standard SS AIS. The proposed methodology keeps the available agents scattered, but still close to the intruders' final position, which is beneficiary in the sense that there is an emerging awareness of the possible future position necessity and this can be exploited to tackle the advantage of the intruders' higher speed capability. This effect comes as a result of the blind follow rule due to the strategy validity threshold, which allows an agent to move towards the last known position of an intruder whose track was recently lost.



(a) (b)

(c) (d)

Fig. 9. Final swarm positions of (a) SS AIS, (b) TE-SS AIS, (c) ED-VS AIS, and (d) PSO algorithms for Test Case 6.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2020.106135.

References

[1] A.S. Tanenbaum, M. van Steen, Distributed Systems: Principles and Paradigms, Prentice Hall, 2002.

[2] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural To Artificial Systems, Oxford University Press, 1999.

[3] H. Hamann, Swarm Robotics: A Formal Approach, Springer International Publishing, 2018.

[4] I. Sommerville, Software Engineering, Pearson, 2015.

[5] G.C. Goodwin, S.F. Graebe, M.E. Salgado, Control System Design, Prentice Hall, 2001.

[6] D.D. Siljak, Decentralized Control of Complex Systems, Dover Publications, 2011.

[7] X.-J. Li, X.-X. Ren, G.-H. Yang, Backstepping-based decentralized tracking control for a class of interconnected stochastic nonlinear systems coupled via a directed graph, Inform. Sci. 477 (2019) 302–320.

[8] B.V. Patil, L.P.M.I. Sampath, A. Krishnan, F.Y.S. Eddy, Decentralized nonlinear model predictive control of a multimachine power system, Int. J. Elec. Power 106 (2019) 358–372.

[9] S. Xu, Z. Yan, D. Feng, X. Zhao, Decentralized charging control strategy of the electric vehicle aggregator based on augmented Lagrangian method, Int. J. Elec. Power 104 (2019) 673–679.

[10] M. Lopez-Franco, E.N. Sanchez, A.Y. Alanis, C. Lopez-Franco, N. Arana-Daniel, Decentralized control for stabilization of nonlinear multi-agent systems using neural inverse optimal control, Neurocomputing 168 (2015) 81–91.

[11] Y. Yang, A. Minai, M. Polycarpou, Decentralized cooperative search in UAV's using opportunistic learning, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, American Institute of Aeronautics and Astronautics, 2002.

[12] A. Sinha, A. Tsourdos, B. White, Multi UAV coordination for tracking the dispersion of a contaminant cloud in an urban region, Eur. J. Control 15 (2009) 441–448.

[13] M. Defoort, T. Floquet, W. Perruquetti, A. Kokosy, J. Palos, 25 - A decentralized planning architecture for a swarm of mobile robots, in: Using Robots in Hazardous Environments, Woodhead Publishing, 2011, pp. 575–590.

[14] S. Selvakumaran, S. Parthasarathy, R. Karthigaivel, V. Rajasekaran, Optimal decentralized load frequency control in a parallel AC-DC interconnected power system through HVDC link using PSO algorithm, Enrgy Proced. 14 (2012) 1849–1854.

[15] R. Hemmati, N. Azizi, M. Shafie-khah, J.P.S. Catalão, Decentralized frequency-voltage control and stability enhancement of standalone wind turbine-load-battery, Int. J. Elec. Power 102 (2018) 1–10.

[16] J. de Lope, D. Maravall, Y. Quiñonez, Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems, Neurocomputing 163 (2015) 47–55.

[17] M. Bereta, T. Burczyński, Immune k-means and negative selection algorithms for data analysis, Inform. Sci. 179 (2009) 1407–1425.

[18] A. Szabo, L.N.d. Castro, M.R. Delgado, FaiNet: An immune algorithm for fuzzy clustering, in: IEEE International Conference on Fuzzy Systems, 2012, pp. 1–9.

[19] B. Schmidt, A. Al-Fuqaha, A. Gupta, D. Kountanis, Optimizing an artificial immune system algorithm in support of flow-based internet traffic classification, Appl. Soft Comput. 54 (2017) 1–22.

[20] M. Navarro, M. Caetano, G. Bernardes, L.N. de Castro, J.M. Corchado, Automatic generation of chord progressions with an artificial immune system, in: International Conference on Evolutionary and Biologically Inspired Music and Art, EvoMUSART 2015, Springer International Publishing, Cham, 2015, pp. 175–186.

[21] G.B. Bezerra, T.V. Barra, H.M. Ferreira, H. Knidel, L.N. de Castro, F.J. Von Zuben, An immunological filter for spam, in: International Conference on Artificial Immune Systems, ICARIS 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 446–458.

[22] J. Twycross, U. Aickelin, A. Whitbrook, Detecting anomalous process behaviour using second generation artificial immune systems, Int. J. Unconv. Comput. (2010).

[23] J. Kim, P.J. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, J. Twycross, Immune system approaches to intrusion detection – a review, Nat. Comput. 6 (2007) 413–466.

[24] A.M. Whitbrook, U. Aickelin, J.M. Garibaldi, Two-timescale learning using idiotypic behaviour mediation for a navigating mobile robot, Appl. Soft Comput. 10 (2010) 876–887.

[25] H.Y.K. Lau, V.W.K. Wong, An immunity approach to strategic behavioral control, Eng. App.l Artif. Intel. 20 (2007) 289–306.

[26] A. Raza, B.R. Fernández, A multi-tier immuno-inspired framework for heterogeneous mobile robotic systems, Appl. Soft Comput. 71 (2018) 333–352.

[27] S. Sang-Joon, L. Dong-Wook, S. Kwee-Bo, Artificial immune-based swarm behaviors of distributed autonomous robotic systems, in: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), vol. 3994, 2001, pp. 3993–3998.

[28] L. Dong-Wook, S. Kwee-Bo, Artificial immune network-based cooperative control in collective autonomous mobile robots, in: Proceedings 6th IEEE International Workshop on Robot and Human Communication, RO-MAN'97 SENDAI, 1997, pp. 58–63.

[29] J. Jin-Hyung, L. Dong-Wook, S. Kwee-Bo, Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system, in: IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028), vol. 616, 1999, pp. 614–619.

[30] J. Guo, G. Hu, H. Wang, Ian-based cooperative control model for multi-agent system, 2011.

[31] L. Weng, M. Xia, K. Hu, Z. Qiao, A memory/immunology-based control approach with applications to multiple spacecraft formation flying, 2013.

[32] L. Weng, Q. Liu, M. Xia, Y.D. Song, Immune network-based swarm intelligence and its application to unmanned aerial vehicle (UAV) swarm coordination, Neurocomputing 125 (2014) 134–141.

[33] A.K. Abbas, A.H. Lichtman, Basic Immunology: Functions and Disorders of the Immune System, W. B. Saunders, 2001.

[34] N.K. Jerne, The generative grammar of the immune system, EMBO J. 4 (1985) 847–852.

[35] N.K. Jerne, Idiotypic networks and other preconceived ideas, Immunol. Rev. 79 (1984) 5–24.

[36] A.S. Perelson, Immune network theory, Immunol. Rev. 110 (1989) 5–36.

[37] J.D. Farmer, N.H. Packard, A.S. Perelson, The immune system, adaptation, and machine learning, Physica D 22 (1986) 187–204.

[38] J.D. Farmer, S.A. Kauffman, N.H. Packard, A.S. Perelson, Adaptive dynamic networks as models for the immune system and autocatalytic sets, Ann. NY Acad. Sci 504 (1987) 118–131.

[39] J. Daudi, An overview of application of artificial immune system in swarm robotic systems, Adv. Robot. Automat. 4 (2015) 127–132.

[40] G.M. Fricke, J.P. Hecker, J.L. Cannon, M.E. Moses, Immune-inspired search strategies for robot swarms, Robotica 34 (2016) 1791–1810.

[41] J. Timmis, P. Andrews, E. Hart, On artificial immune systems and swarm intelligence, Swarm Intel-US 4 (2010) 247–273.

[42] A. Mondal, A. Ghosh, S. Ghosh, Scaled and oriented object tracking using ensemble of multilayer perceptrons, Appl. Soft Comput. 73 (2018) 1081–1094.

[43] W. Li, Z. Dou, L. Qi, C. Shi, Wavelet transform based modulation classification for 5g and UAV communication in multipath fading channel, Phys. Commun.-AMST (2019).

[44] A. Raza, B.R. Fernandez, Immuno-inspired robotic applications: A review, Appl. Soft Comput. 37 (2015) 490–505.

[45] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.

[46] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), vol. 1672, 2002, pp. 1671–1676.

[47] F. Buonamici, M. Carfagni, R. Furferi, L. Governi, A. Lapini, Y. Volpe, Reverse engineering of mechanical parts: A template-based approach, J. Comput. Des. Eng. 5 (2018) 145–159.

[48] S. Spanogianopoulos, Q. Zhang, S. Spurgeon, Fast formation of swarm of UAVs in congested urban environment, IFAC PapersOnLine 50 (2017) 8031–8036.

[49] G. Oh, Y. Kim, J. Ahn, H.-L. Choi, PSO-based optimal task allocation for cooperative timing missions, IFAC PapersOnLine 49 (2016) 314–319.

[50] A. Belkadi, H. Abaunza, L. Ciarletta, P. Castillo, D. Theilliol, Distributed path planning for controlling a fleet of UAVs: Application to a team of quadrotors, IFAC PapersOnLine 50 (2017) 15983–15989.