Dynamic UAV Swarm Deployment for Non-Uniform Coverage

Robotics Track

Dario Albani
ISTC-CNR, Sapienza University of Rome
Rome, Italy
dario.albani@istc.cnr.it

Tiziano Manoni
Sapienza University of Rome
Rome, Italy
manoni.1314894@studenti.uniroma1.it

Daniele Nardi
Sapienza University of Rome
Rome, Italy
nardi@diag.uniroma1.it

Vito Trianni
ISTC-CNR
Rome, Italy
vito.trianni@istc.cnr.it

ABSTRACT

In many monitoring and mapping applications, high-resolution data are required only in certain areas while others can receive lower attention. To this end, unmanned aerial vehicles (UAVs) can adjust the flight altitude to increase the resolution only where needed, making non-uniform coverage strategies efficient both in time and energy expenditure. In a multi-UAV monitoring context, it is necessary to deploy UAVs to inspect in parallel those areas where a higher resolution is required. To address this problem, we propose a decentralised deployment strategy inspired by the collective behaviour of honeybees. This strategy dynamically assigns UAVs to different areas to be monitored, and suitably re-assigns them to other areas when needed. We introduce an analytical macroscopic model of area monitoring from UAVs, and we propose a parameterisation that leads to an efficient allocation of UAVs to the areas to be monitored. We exploit abstract multi-agent simulations to study the dynamics of the deployment of UAVs to multiple areas, and we present results with simulations of a UAV swarm engaged in a weed monitoring and mapping task.

KEYWORDS

UAV swarms; deployment; area coverage; precision agriculture.

ACM Reference Format:
Dario Albani, Tiziano Manoni, Daniele Nardi, and Vito Trianni. 2018. Dynamic UAV Swarm Deployment for Non-Uniform Coverage. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

Unmanned aerial vehicles (UAVs) are powerful tools for several applications where aerial monitoring is required [27, 28]. Often, the area to be inspected presents a heterogeneous distribution of relevant features, granting a significant efficiency advantage to non-uniform coverage strategies [34, 35]. This is the case, of instance, for weed mapping problems in precision agriculture [29]. Weeds often grow in patches, making it advantageous to closely monitor only those areas where their density is high, while only mildly inspecting areas devoid of weeds. However, despite the potential advantage of similar strategies, the state of the art in UAV monitoring for agricultural applications counts mostly uniform coverage approaches [12, 29, 39, 42], in which navigation follows a standard "lawnmower" path [13].

Non-uniform coverage strategies for UAVs assume that a coarse estimation of the monitoring effort can be obtained by UAVs flying at high altitude, so that images of the area are obtained at a somewhat low resolution, however sufficient to get hints about the requirements of that region. By flying at lower altitudes, inspection can be conducted with a higher resolution only in correspondence of interesting areas, resulting in a substantial saving in the coverage time and effort [34, 35]. Additional advantages can be obtained by adopting a multi-robot approach, which offers the possibility to parallelise operations and exploit collaboration to cover large areas in a short time [10, 16]. A straightforward solution consists in decomposing the field in non-overlapping areas to be assigned to different UAVs [2, 25]. Moreover, collaboration can be further exploited to improve the mapping quality, especially when individual inspection is error-prone: by re-sampling the same area at different times and from different perspectives, multi-robot systems can achieve higher accuracy in the generated map [1]. Such a collaborative approach requires that a number of UAVs are deployed to the same area and collaborate to the monitoring task. Provided that different areas may have different requirements in terms of monitoring effort, as postulated by non-uniform coverage approaches, the problem of determining where and when UAVs should be deployed requires particular attention.

Background. Apart from the details related to the navigation of UAVs over the field, such a deployment problem can be related to multi-robot task allocation (MRTA), and specifically to the single-task robot, multi-robot task, instantaneous assignment category (ST-MR-IA) [14], whereby each robot can execute just a single task at the time without planning requirements, but multiple robots can collaborate to bring forth a given task. Several approaches have been proposed in the literature for ST-MR-IA problems, especially for the related coalition formation problem in multi-agent systems and robots [41], often resorting to market-based approaches, which however are demanding in terms of communications, and are known to scale poorly with the number of agents and tasks [8]. Threshold-based approaches [5, 19, 20, 22] or stochastic task switching [3, 7, 15] are more suitable for decentralised task allocation. The former are based on the well-known response thresholds model,

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

whereby agents engage in a task if some task-related stimulus overcomes a given internal threshold [4]. Correctly dimensioning the individual thresholds to obtain a desired collective response and to deal with variability in the number of tasks and robots available is however difficult, and has led to the introduction of adaptive thresholds that implement a simple form of individual learning [5, 20, 22]. Stochastic task switching approaches have very low requirements in terms of individual decision-making abilities, as engaging in a task is determined by constant switching rates, which need however to be optimised to achieve a desired allocation [3, 7, 15, 21, 26]. Such optimisation is performed beforehand on the basis of the available knowledge, or during task execution by a central authority that oversees the process. To bring forth such optimisation process, several methods have been proposed based on macroscopic dynamical models, which have also been extended to take into account the spatial distribution of tasks and robots [11, 43]. In this work, we advance beyond these studies by proposing a fully decentralised strategy that exploits simple broadcast communication among UAVs to support coordination and collaboration on the available tasks. Differently from previous approaches, the proposed strategy has no central authority with full information about the task progress. Also, the deployment requirements are not known a priori and vary through time as the mapping is accomplished. For this reason, we employ a strategy based on communication and collaboration among UAVs, which interact to determine the best deployment in each area according to the current estimated utility.

Problem and approach. We consider a swarm of UAVs (hereafter also agents) that implement a non-uniform coverage strategy for weed mapping [1, 29]. The swarm is composed of N agents that collaboratively monitor a field. The field is partitioned in M non-overlapping areas, and each area A_i is characterised by a weed density W_i. Each area is further divided into M_c smaller cells for low-altitude/high-resolution inspection, and each cell c is characterised by a local weed density w_c. We define the *utility* $u_i(t)$ of an area as a function of the number of cells that require close inspection and mapping, weighted by the local weed density w_c. The initial value $u_i(0)$ can be estimated from W_i as obtained through high-altitude/low-resolution inspection. UAVs monitor an area exploiting a robust and scalable algorithm [1], and communicate with neighbours within a distance d to share information about the inspected cells. UAVs adopt a flooding routing strategy [23] to re-broadcast packets and improve information spreading within the network (see Section 2.3.2 for details). In this way, the utility value $u_i(t)$ is locally known by each UAV and is updated upon message reception to reflect the need for additional monitoring. An agent considers an area as completely mapped when the residual utility is lower than a low threshold $u_i < \nu$. Mapping of the i^{th} area terminates at completion time τ_i, i.e. when all agents know that its utility is lower than the given threshold. The target non-uniform coverage strategy must therefore allow the UAVs to identify the areas whose utility is above threshold, and to deploy and (re-)deploy a sufficient number of UAVs in each area to minimise the total completion time $\tau_M = \max_i \tau_i$.

The design of a decentralised strategy for non-uniform coverage—discussed in Section 2—is performed with a top down approach that starts from an abstract model of the variation of the utility of an area under a given inspection effort by a group of UAVs (see Section 2.1). We then design the decentralised strategy starting from a macroscopic model describing the UAVs deployment dynamics in terms of rate equations, similarly to previous work developed in the context of robot swarm deployment and task allocation [3, 7, 15]. By communicating with each other, UAVs can collectively decide to converge towards areas of high utility, can quickly abandon areas with no (residual) utility, or can distribute over multiple areas to act in parallel. Drawing inspiration from the behaviour of honeybees [17, 32, 36], we implement a decentralised strategy after a design pattern for collective decisions [32, 33], extending it to adress MRTA problems and adapting it to variable working conditions in terms of number of areas to inspect, area utility and number of available UAVs. In Section 2.2, we introduce a coupling between the macroscopic model of the target UAV deployment dynamic and the model describing the utility dynamics. We derive a parameterisation for the coupled model to maximise the collective inspection efficiency and, finally, from the macroscopic model we derive a microscopic description of the individual UAV behaviour including the low-altitude/high-resolution inspection strategy (see Section 2.3).

The obtained results are presented in Section 3, where we verify the link between macroscopic and microscopic descriptions through abstract multi-agent simulations (see Section 3.1), and we proceed to study the system efficiency in more complex working conditions entailing multiple areas to be monitored (see Section 3.2). Finally, we evaluate the dynamic deployment strategy through simulations of a UAV swarm engaged in a weed monitoring and mapping task, in which UAVs communicate locally with neighbours (see Section 3.3). Here, we consider the spatial distribution of the areas to be monitored, and we account for the need of high-altitude/low-resolution inspection prior to deployment, given that the actual deployment requirements are not known at start. Despite we not explicitly consider spatiality in our macroscopic modeling [11, 43], we demonstrate good agreement with the non-spatial macroscopic models, although dependent on the ability of UAVs to spread information widely through the local communication network. In particular, we show how the chosen parameterisation allows switching from a utility-proportional deployment, to a winner-take-all allocation in which the areas with higher utility are inspected with priority over the other areas, to an utility responsive deployment in which agents immediately reacts to changes in areas utilities. In this latter case, we observe an adaptive behaviour of the UAVs, that balances the mapping needs of different areas with the available resources. In Section 4, we discuss the merits, limitations and challenges of the proposed approach, together with possible future developments.

2 SYSTEM DESIGN

2.1 Macroscopic model of the utility dynamics

The problem so far described can be abstracted to focus on the deployment aspects and their link with the variation of the utility over time. To this end, we introduce a simple and general macroscopic model of the utility dynamics. We partition the population of agents into a group of n_u agents performing high-altitude/low-resolution monitoring, and M groups of n_i agents allocated to area A_i (i.e., performing low-altitude/high-resolution inspection) and, so that

$N = n_u + \sum_i n_i$. The variation of utility of area A_i depends only on the collaborative effort of the n_i allocated agents, as follows:

$$\dot{u}_i = -u_i n_i(\delta n_i - \xi n_i^2), \quad u_i \in [0,1]. \tag{1}$$

Here, the utility variation depends on the current value u_i resulting in an exponential decay, as well as on the contribution from the n_i allocated agents. The parameters δ and ξ are rates of change that represent, respectively, the collaborative contribution of allocated agents to the inspection task, and the effects of interferences between agents. Here, we assume that the latter are caused by harmful interactions—e.g., due to overcrowding—which are of higher order than the beneficial interactions resulting from collaboration. To obtain a monotonically decreasing utility, we also assume that $\xi < \delta$ and that $n_i \leq \delta/\xi$. In this work, we consider δ and ξ as constant extrinsic parameters related to the coverage strategy within an area, which can be estimated as done in Section 3.3.

Although somewhat arbitrary with respect to a realistic scenario, this model of the utility dynamics is general enough to accommodate features that are relevant for any non-uniform coverage task and it has been validated against the UAV simulations, as discussed in Section 3.3. The model includes the effects of the individual low-altitude/high-resolution inspection, as well as collaboration and interferences among agents. Specifically, it provides a mean to take into account the beneficial effects of collaboration before interferences take over, as shown by the analytical derivations provided in the supplementary material. By solving the ODE in (1) for a fixed n_i, it is possible to derive the value $n^\star$ that minimises the completion time τ_i, as detailed in the supplementary material:

$$n^\star = \frac{2\delta}{3\xi}. \tag{2}$$

Interestingly, the optimal number of agents to be deployed in each area does not depend on the initial utility $u_i(0)$, but the completion time τ_i does. This knowledge can be exploited to design a strategy that can help minimising the overall completion time.

2.2 Macroscopic model of agent deployment

As mentioned above, UAVs need to identify the areas of interest and estimate their initial utility. On such a basis, they have to choose an area to inspect, possibly in collaboration with other agents so as to minimise the completion time. The problem contains elements of collective decisions—when agents need to converge to a single area to inspect—and task allocation—when agents need to disperse on different areas, and can be suitably tackled following a design pattern inspired by the honeybee nest-site selection behaviour [33]. The design pattern defines a macroscopic model of the system dynamics, which—in its most general form [32]—reads as follows:

$$\begin{cases} \dot{x}_i = \gamma_i x_u - \alpha_i x_i + \rho_i x_u x_i - \sum_{j=1}^{M} x_j \beta_{ji} x_i \\ x_u = 1 - \sum_{i=1}^{M} x_i, \qquad \gamma_i, \alpha_i, \rho_i, \beta_{ij} \geq 0 \end{cases} \tag{3}$$

where $x_i = n_i/N$ represents the fraction of agents deployed on area A_i (also referred to as "committed" agents [33]), and x_u represents the fraction of "uncommitted" agents, that is, agents available for deployment in any area. The macroscopic dynamics are determined by four concurrent processes: (i) uncommitted agents spontaneously enroll to area A_i at rate γ_i; (ii) committed agents spontaneously abandon an area A_i at rate α_i; (iii) agents committed to area A_i can recruit uncommitted agents at rate ρ_i; (iv) agents committed to area A_j "inhibit" agents committed to A_i at rate β_{ji}, so that the latter become uncommitted. Differently from previous studies, we extend here the inhibition paradigm to act between any pair of committed agents, hence including both cross-inhibition ($\beta_{ji} \neq 0$ for each $j \neq i$) and self-inhibition ($\psi_i \triangleq \beta_{ii} \neq 0$). As we will discuss below, self-inhibition allows to finely control the number of agents working in the same area.

2.2.1 Model of decentralised deployment in a single area. When only one area requires inspection, agents need to be deployed to match the optimal number $n^\star$ as defined in (2). The ODE system in (3) can be simplified for the single area case ($M = 1$, hence $x_u = 1 - x_i$) as follows :

$$\dot{x}_i = \gamma_i(1 - x_i) - \alpha_i x_i + \rho_i x_i(1 - x_i) - \psi_i x_i^2 \tag{4}$$

This ODE can be easily solved to find that, under valid parameterisations, it always presents a stable fixed point $x^\star$ (see the supplementary material). By requiring that $Nx^\star = n^\star$ and solving for ψ_i in the special case of $\alpha_i = 0$ (i.e., no spontaneous abandonment of area A_i, see below for a discussion on this parameter), we obtain the following condition on the system parameters:

$$\psi_i = \frac{(3\xi N - 2\delta)(3\xi N \gamma_i + 2\delta\rho_i)}{4\delta^2}, \quad N \geq \frac{2\delta}{3\xi} = n^\star \tag{5}$$

This means that whenever the number of available agents N is higher than the optimal value, self-inhibition should take place at a rate ψ_i from (5) to compensate spontaneous enrollment (at rate γ_i) and recruitment (at rate ρ_i).

2.2.2 Choice of the macroscopic parameterisation. Following the design pattern guidelines [33], a suitable parameterisation must be defined which provides desired macroscopic properties. To this end, we exploit the results presented in [32], which introduce a utility-dependent parameterisation that brings agents to consensus on a single area only if it has a high utility. In our case, considering the dynamics of the area utility as in (1), we define the parameters determining the macroscopic behaviour as follows:

$$\gamma_i = k u_i \tag{6}$$

$$\alpha_i = k\mathcal{H}(\nu - u_i) \tag{7}$$

$$\rho_i = h u_i \tag{8}$$

$$\beta_{ij} = \mathcal{R}(h u_i \frac{2\delta - 3\xi n_j}{2\delta}), \quad i \neq j \tag{9}$$

where $\mathcal{H}(\cdot)$ is the Heaviside step function and $\mathcal{R}(\cdot)$ is the ramp function. Here, spontaneous enrollment and recruitment are directly proportional to the residual utility u_i. Spontaneous abandonment is defined in order to take place only when an area is completely mapped, hence the residual utility is lower than the threshold ν (hence justifying the choice of null abandonment to obtain (5)). Conversely, we define cross-inhibition proportionally to the area utility, as in [32], but we discount the rate by a function of the number of enrolled agents n_j, so that cross-inhibition becomes less relevant the more the number of committed agents approaches the optimum $n^\star$. The rationale is that cross-inhibition should not take

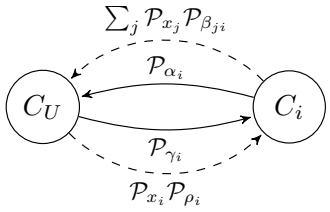

Figure 1: PFSM implementing the individual behaviour. C_i represents the state for an agent committed to a generic area A_i, while C_u represents the uncommitted agent. Solid and dashed arrows represent spontaneous and interactive transitions, respectively. For simplicity, we omit here to represent auto-loops and the corresponding probabilities.

place when an area has a near-optimal number of deployed agents. Finally, we define the self-inhibition rate as follows:

$$\psi_i = \mathcal{R}\left(\frac{(3\xi\hat{N} - 2\delta)(3\xi\hat{N}\gamma_i + 2\delta\rho_i)}{4\delta^2}\right), \quad \hat{N} = n_i + n_u \tag{10}$$

which differs from (5) only in the usage of $\hat{N}$ instead of N, as self-inhibition needs to compensate the actual spontaneous enrollment and recruitment rates, which depend on n_u and n_i respectively. The ramp function here ensures that the rate is non-negative, hence respecting the condition $\hat{N} \geq n^\star$ from (5).

2.3 Microscopic implementation

Once the macroscopic parameterisation has been defined, it is possible to implement a multi-agent system following the guidelines provided by the design pattern [33], and then translate the implementation for the UAV simulations. The multi-agent simulations presented in Section 2.3.1 represent an abstract deployment scenario in which agents are not distributed in space. Agents belong to an abstract well-mixed population where interactions take place randomly, hence simplifying issues related to inter-agent communication. As a result, there is no difference in the interaction probability between committed and uncommitted agents. The UAV simulation described in Section 2.3.2 include instead agents moving in space and interacting within a limited communication range. These simulations are useful to prove that the design methodology works against realistic conditions.

2.3.1 Multi-agent simulations. Agents are initially provided with the list of the M areas A_i together with their initial utility $u_i(0)$. Each agent implements a probabilistic finite state machine (PFSM) with M+1 states, one for being uncommitted and M for commitment to each area A_i (see a compact representation in Figure 1 where we represent only one state for the commitment to the generic area A_i). The transition probabilities $\mathcal{P}_\lambda$—with $\lambda \in \{\gamma_i, \alpha_i, \rho_i, \beta_{ij}\}$, see Figure 1—are computed according to the design pattern [33], and substituting (6) to (10), as follows:

$$\mathcal{P}_\lambda = \lambda\Delta t, \quad \lambda \in \{\gamma_i, \alpha_i, \rho_i, \beta_{ij}\} \tag{11}$$

where Δt is the constant update time for the PFSM which must be correctly dimensioned to obtain valid transition probabilities for the PFSM (i.e., all the exit transition probabilities, including the probability to remain in the same state, sum to 1). Here, we consider $\Delta t = 0.1s$. The interactive transitions are executed upon interaction with another agent selected at random, on the basis of its commitment state to area A_i, which happen with probability $\mathcal{P}_{x_i}$. We assume that parameters present in (6)–(10) can be estimated a priori (e.g., δ and ξ), or on the basis of messages exchanged among agents about their commitment state (e.g., n_i, n_u).

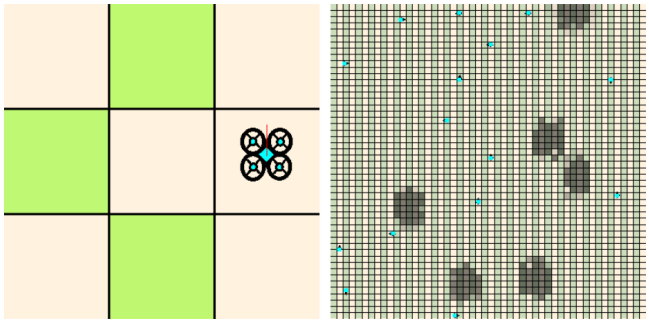

Figure 2: The simulation environment. (Left) The high-altitude partition of the field in $M = 9$ areas, green corresponding to areas where weeds are present. (Right) The low-altitude partition of an area in $M_c = 50 \times 50$ cells, grey zones represent weed patches and blue dots represent drones.

2.3.2 UAV swarm simulations. A multi-agent simulation of UAVs monitoring and mapping weeds in a field has been developed using the MASON simulation environment [24].[1] We have implemented a simulated environment following the problem description given in Section 1, so that the environment is divided in M areas, each further divided into M_c square cells of 1 m side. Within an area A_i, weeds can be found in patches, with a number p_i of patches per area, each patch extending over several cells (see Figure 2). Differently from previous studies [1], here we focus on the deployment problem and we abstract other relevant factors for UAV swarm monitoring and mapping such as sensor or state estimation error.

UAVs move at constant speed v, and implement the PFSM in Figure 1 to determine which area to inspect. When uncommitted, UAVs move between areas following an isotropic, uncorrelated random walk [9]. UAVs choose the next area to visit at random among the four nearest areas in the grid, hence minimising the travel time between visiting an area and the following one. This ensures that the UAVs spread uniformly over the areas and that each area has a roughly constant probability of being visited. Whenever a UAV reaches a new area and acquires a position from which it can be completely observed, it estimates W_i and therefore u_i, and broadcasts this information to the other UAVs to inform them about the existence, location and utility of areas to be monitored. At every control step, UAVs also broadcast their position and state, that is, whether uncommitted or committed to area A_i, together with the residual utility u_i of the area they are committed to. Upon reception of a new message (recognised through a unique identifier made of

[1] In this work, we neglect collisions among UAVs and assume that they implement a robust collaborative avoidance strategy based on the velocity obstacle approach [37].

source ID and time-stamp), the UAV re-broadcasts it to its neighbours, implementing a flooding routing protocol[2] that allows to quickly spread the message to the swarm [1, 23]. The efficiency in spreading information depends on the maximum communication distance d and the spatial distribution of the UAVs. If the resulting communication network is connected, broadcasted messages can reach any UAV. Otherwise, some UAV may miss the message. Exploiting communication, each UAV maintains a list $\mathcal{L}_A$ of the areas to be monitored and updates their utility on the basis of the progress in monitoring by committed UAVs. Also, each UAV maintains a list $\mathcal{L}_U$ of the known UAVs and their current state using the latest received messages.

On the basis of the residual utilities of the known areas A_i in $\mathcal{L}_A$, uncommitted UAVs spontaneously enroll with probability $\mathcal{P}_{\gamma_i}$. Additionally, at each decision step, UAVs choose another known agent from $\mathcal{L}_U$ at random and possibly get recruited with probability $\mathcal{P}_{\rho_i}$ if the latter is committed to area A_i. Whenever an uncommitted UAV becomes committed to a given area, it flies over the area and then descends to the low-altitude layer in a random position. When committed, UAVs move to the low-altitude layer and start monitoring the field in collaboration with others, focusing on cells that have not been covered [1]. UAVs implement a stochastic exploration strategy based on a reinforced random walk [6, 9], thereby following a correlated random walk and interacting with neighbours to avoid interferences. A mechanism to avoid returning on previously visited cells leads to a very efficient coverage, although stochastic. Communication among UAVs is exploited to share information so as to avoid re-visiting areas that have been already visited before. Also, UAVs exploit knowledge of the others position to maximise separation and increase the coverage efficiency. For more details about the collaborative strategy for low-altitude/high-resolution inspection, see [1]. Following the PFSM in Figure 1, UAVs committed to area A_i may become uncommitted either through spontaneous abandonment with probability $\mathcal{P}_{\alpha_i}$, or when randomly choosing another UAV from $\mathcal{L}_U$ that is committed to area A_j through some inhibition mechanism (self/cross), with probability $\mathcal{P}_{\beta_{ji}}$.

3 RESULTS

3.1 Micro-macro link for simple deployment

As a starting point, we study a coupled ODE system for the simplest case—deployment within a single area—putting together (1) and (4), and exploiting the chosen parameterisation as in (6)–(10). This allows to verify the link between microscopic implementation and macroscopic description. The dynamics of the macroscopic model are displayed in Figure 3 left (for additional details, see the supplementary material). It is possible to note that the dynamics converge to the optimal value $x^\star = n^\star/N$, with a speed that is determined by h and k (having a rather low value in Figure 3 to highlight the dynamics). Then, x_i remains constant and the utility u_i decreases at maximum speed, until the threshold ν is reached. At this time, abandonment becomes non null and dominates the system dynamics, leading to a quick convergence to the stable fixed point (black dot). Once we verified the correctness of the macroscopic dynamics, we look at the adherence of the multi-agent implementation with the model predictions. The right panel in Figure 3 shows the time evolution of u_i and x_i for the same conditions discussed above. Besides the insights in the speed of the process, the figure shows the precise match obtained between macroscopic model and multi-agent simulations, confirming the correctness of the implementation.

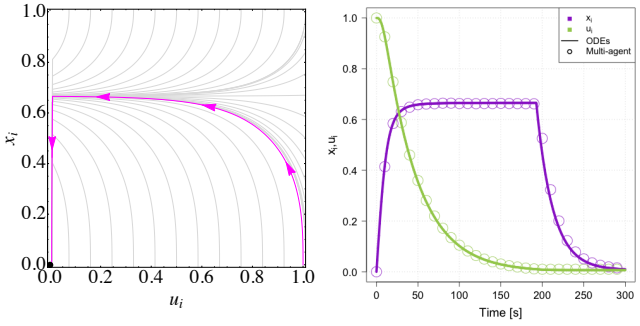

Figure 3: Dynamics of the coupled system for single area deployment ($M = 1$, $N = 75$, $k = h = 0.005$, $\delta = 3\text{E}{-6}$, $\xi = 4\text{E}{-8}$, corresponding to $x^\star = \frac{2}{3}$). Left: trajectories of the macroscopic model in the phase space $\langle x_i, u_i \rangle$. The bold magenta line highlights the trajectory starting from the point $u_i(0) = 1$ and $x_i(0) = 0$, meaning that the utility is at its maximum and all agents are initially uncommitted. Grey lines show trajectories for additional starting points. Right: Micro-macro link between the ODE system and the multiagent simulations.

3.2 Deployment dynamics with multiple areas

The decentralised deployment problem becomes more interesting and challenging as soon as the number of areas increases. We consider here multi-agent simulations with the complete system for $M > 1$, hence including also cross-inhibition between agents committed to different areas. Similar to [32], we introduce the control parameter $r = h/k$ which represents the ratio between interactive and spontaneous transitions, to be used to control the system dynamics. To understand how the parameter r affects the deployment in the different areas, we introduce two metrics: the allocation error $\mathcal{E}(t)$, and the completion progress $C(t)$. The former measures the deviation of the number of allocated agents from the optimal value $n^\star$, provided that a sufficient number of agents is available. The latter indicates the level of accomplishment of the inspection tasks, normalised to the overall requirements.

The allocation error is computed as follows:

$$\mathcal{E}(t) = \frac{1}{N} \sum_{i=1}^{M} \left| \hat{n}_i(t) - \min\left(N - \sum_{j=1}^{i-1} \hat{n}_j(t),\ n_i^\star(t) \right) \right|, \qquad (12)$$

where $\hat{n}_i(t)$ represents the deployment in the i^{th} area in decreasing order of allocated agents at time t. Here we consider the utility-dependent optimal deployment for area i that becomes null as soon as the residual utility decreases below the threshold ν:

$$n_i^\star(t) = \begin{cases} n^\star & \text{if } u_i(t) \geq \nu \\ 0 & \text{otherwise} \end{cases} \qquad (13)$$

[2] Despite not being efficient in terms of number of messages exchanged, this routing protocol ensures the maximum reach for each message. To improve communication efficiency, other protocols can be chosen, as discussed in [1].

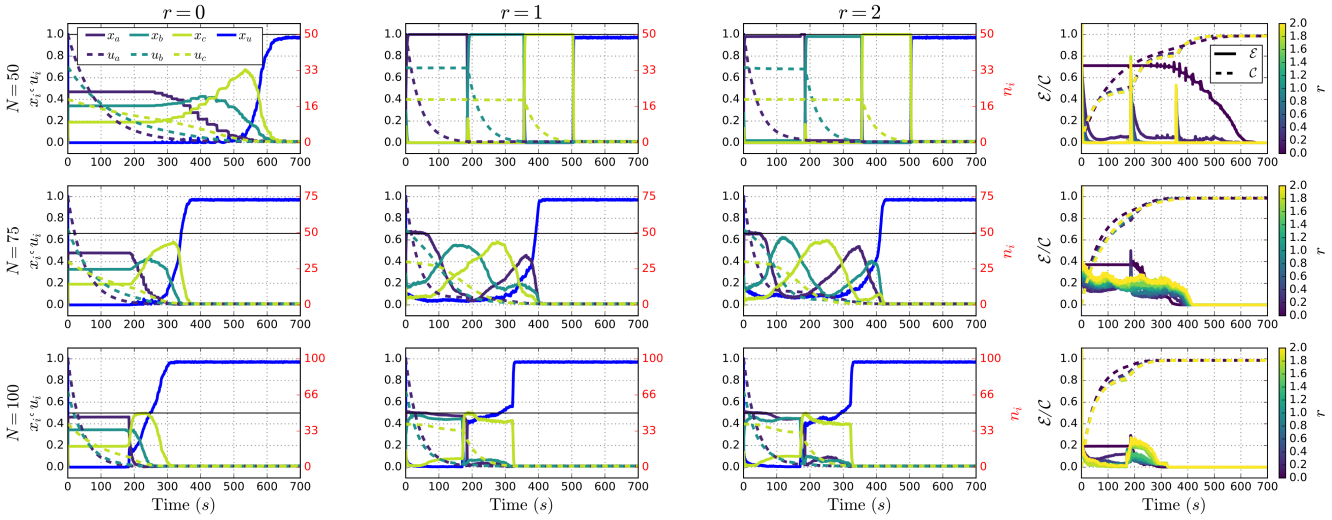

Figure 4: Deployment dynamics with $M = 3$ areas, averaged over 100 simulation runs for $N = 50$ (top row), $N = 75$ (middle row) and $N = 100$ agents (bottom row). Left-Center columns: Temporal dynamics of the utility u_i and the fraction of committed/uncommitted agents (x_i/x_u) for different values of r (respectively: $r = 0$; $r = 1$ and $r = 2$). The black horizontal line indicates the optimal fraction $x^\star = n^\star/N$. Right column: allocation error $\mathcal{E}(t)$ and completion progress $C(t)$ for varying values of r.

In other words, at each time t, the allocation error $\mathcal{E}(t)$ is computed starting from the area with the highest number of allocated agents (i.e., $\hat{n}_1(t)$) and computing the deviance from the optimal value $n_i^\star(t)$, or N should it be lower than the optimum. We repeat for the second highest number of allocated agents $\hat{n}_2(t)$, for which there are now only $N - \hat{n}_1(t)$ available agents, and we check the deviance with respect to $n_2^\star(t)$. The procedure is repeated for all M areas. The allocation error $\mathcal{E}(t)$ is null if the deployment is optimal in all areas, given the number of available agents, but is higher than zero if there is some deviation somewhere. It can be used to evaluate the performance in deployment without any reference to the priority of areas according to their utility. The completion progress $C(t)$ is simply computed according to the residual utility of the different areas:

$$C(t) = \frac{\sum_{i=1}^{M} (u_i(0) - u_i(t))}{\sum_{i=1}^{M} u_i(0)} \tag{14}$$

The completion progress takes values in $[0, 1]$ and indicates the fraction of the overall demand that has been accomplished.

We study the system dynamics with $M = 2$ and $M = 3$ areas requiring inspection. We present a large set of simulation results performed for different values of the ratio r and for groups of different size $N \in \{25, 50, 75, 100\}$. Each experimental condition has been replicated for 100 independent runs. For all these simulations, we set $\delta = 6\text{E}{-6}$ and $\xi = 8\text{E}{-8}$, leading to $n^\star = 50$. In this way, we analyse the deployment dynamics both when the total number of agents is insufficient to reach the optimal value $n^\star$, and when agents are in larger numbers.

To understand the deployment process, we discuss the case for $M = 3$ which presents richer dynamics,[3] as the agents have to decide how to split among the areas, and which one has priority according to their utility. Additionally, as soon as the relative utility changes, it can be useful for some agents to switch areas. We have run experiments with $u_1(0) = 1$, $u_2(0) = 0.7$ and $u_3(0) = 0.4$, hence having a clear sequence in the priority of areas given the initial conditions. We fix $k = 0.2$ and we vary h to explore different ratios $r \in [0, 2]$. Results for different values of N are shown in Figure 4.

When $N \leq n^\star$, there are just enough agents to reach the optimal value in one single area. The dynamics for $N = 50$ are shown in the top row in Figure 4 for $r \in \{0, 1, 2\}$. When $r = 0$, there is no recruitment or cross-inhibition among agents, and the only processes that take place are spontaneous enrollment, self-inhibition and abandonment when the inspection task is completed (i.e., $u_i < \nu$). As a consequence, the initial deployment of agents is proportional to the starting utility values (i.e., $x_i \simeq u_i / \sum_j u_j$). As soon as the first area is completely inspected, agents are re-deployed to the other areas. When $r > 0$, recruitment and cross-inhibition play a crucial role in making all agents converge to the same area, which they choose according to the initial utility. Given that N is just enough to reach the optimal value, there are very few uncommitted agents ($x_u \simeq 0$), and also there is no cross-inhibition or self-inhibition in place, so that the deployment remains stable until the utility drops below the threshold ν. At that point, the whole group is fully re-deployed to the second area. When also this is fully inspected, agents are re-deployed to the third area. Overall, we note that the parameter r controls the deployment strategy, switching from a *utility-proportional deployment* when $r = 0$ to a *winner-take-all* strategy with larger values. The effects of the parameter r are clearly visible in the rightmost panel in Figure 4 (top row), where the allocation error is very large for $r = 0$, and strongly decreases for larger values, becoming null for most of the time apart from deviations during the deployment and re-deployment phases. The effects of the deployment strategy are visible also in the completion

[3] See the supplementary material for analogous results for $M = 2$.

progress, which proceeds through three phases when $r > 0$, overall obtaining a faster completion than for the case with $r = 0$.

The collective dynamics are more interesting when $N > n^\star$, as shown in the second and third row of Figure 4. When $r = 0$, we observe again a utility-proportional deployment, for both $N = 75$ and $N = 100$. As soon as the first area is fully inspected, agents get re-deployed to the second and third areas, with self-inhibition constraining the number of agents within the prescribed bounds (i.e., $n_i < n^\star$ and $x_u > 0$). Note that for $N = 100$, the utility u_2 decreases quickly as many agents are enrolled from start to A_2, and goes largely below the utility u_3. As a consequence, when area A_1 is fully inspected, agents are redeployed directly to A_3 as there is no real need to add forces to the inspection of area A_2. Such a *utility-responsive deployment* is more evident for larger values of r, when recruitment and cross-inhibition kick in. When $r = 1$, the initial deployment favours areas with higher utility, thanks to the combined effect of recruitment and cross-inhibition. For $N = 100$ (bottom row in Figure 4), agents are mostly deployed to areas A_1 and A_2, reaching a value close to $n^\star$. However, the more u_1 and u_2 decrease, the more agents get attracted to area A_3, until a massive re-deployment takes place so that few agents remain in their initial areas to complete the inspection, and the majority focuses on the so-far-neglected area A_3. Similar dynamics characterise also the case for $N = 75$ and $r > 0$, where however we observe a more-pronounced responsiveness of agents to the residual utility of the different areas. Indeed, it is possible to observe that agents get deployed first to the areas with higher utility, and then react to the utility dynamics by moving to those areas that are most attractive. Note that the utility-responsive deployment is not instantaneous, but agents take some time to leave an area in favour of another one, which depends on the timescale of the deployment dynamics.

Overall, recruitment and cross-inhibition lead to a competition between different areas to attract agents, and higher values of r lead to a more dynamic deployment and re-deployment of agents. This dynamical condition is reflected in the allocation error (Figure 4, rightmost panels). Here, the utility-proportional deployment presents the highest error in the initial phases, and improves as soon as agents complete the inspection in some areas. The allocation error decreases with increasing values of r until a optimal value is reached, to increase again for larger values. This is mainly the effect of the stronger cross-inhibition, which leads to a higher presence of uncommitted agents ($x_u > 0$), especially for $N = 75$, causing a sub-optimal allocation. In the later stages, the error is also caused by areas with a low residual utility, which are not attractive enough to reach the optimal value $n^\star$. Nevertheless, the error values are reasonably small, representing a cost to be payed for the utility-responsive deployment of agents (see also the discussion in Section 4). Such a cost also corresponds to a slightly slower completion progress with respect to the utility-proportional deployment, as shown in Figure 4.

Finally, Figure 5 shows how the completion time τ_M scales with the number of agents N and varying r (solid lines for $M = 2$, dashed lines for $M = 3$). As expected, the availability of additional agents is beneficial and the completion times decrease significantly. We also note that when $N \leq n^\star = 50$, higher values of r lead to a significant improvement of the completion time with respect to the case with $r = 0$. In these conditions, agents perform better if they converge towards the same area in order to collaborate and be more efficient. When $N > n^\star$, instead, performance slightly degrades with increasing r—although barely noticeable—for the effects of the utility-responsive deployment discussed above.

3.3 Dynamic deployment of UAV swarms

Simulations have been performed to study a more realistic problem whereby UAVs have to move to different areas and actually detect the presence of weeds. Here, the spatial distribution of areas that require inspection—not considered in the results presented above—may play a major role in determining the deployment dynamics [30, 31]. We consider here $M = 9$ areas distributed over a 3×3 grid, each divided in $M_c = 2500$ cells on a 50×50 grid. Each cell is $1\,\mathrm{m}^2$, and UAVs fly at cruise speed of $v = 1\,\mathrm{m/s}$. To compare with previous results, we impose that only three areas have a non null initial utility. Within these areas—randomly selected and labeled as A_i, with $i = 1, 2, 3$—we distribute respectively $p_1 = 6$, $p_2 = 4$ and $p_3 = 2$ weed patches, hence having a different initial utility for each one, normalised in $[0, 1]$. Note that UAVs need to discover these areas among the 9 available before becoming committed.

To implement the devised strategy, we first estimate the value of δ and ξ for the UAV simulation, by using a single area with $p = 6$ weed patches, and by fixing $n_i \in [1, 50]$. We record the variation of the utility over time, and we then fit the function obtained from integrating (1) to estimate the parameters δ and ξ through a least squares method. We obtain a reasonably good fit—although not perfect—that results in $\delta \simeq 0.053$ and $\xi \simeq 0.002$, corresponding to $n^\star \simeq 18$ according to (2).

We run simulations with $N \in \{9, 18, 27, 36\}$ UAVs with communication distance $d \in \{50, 100, 150\}$ m, and we compare with the ideal case of a fully connected network ($d = \infty$). The dynamics displayed by the system correspond to those presented for the abstract multi-agent simulations. However, due to the random spatial distribution, areas with low utility may be discovered first and may quickly

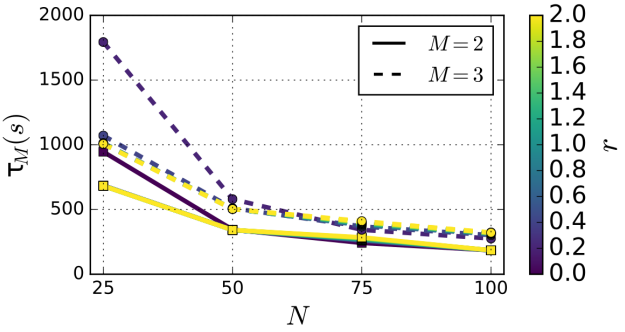

Figure 5: Total completion time τ_M as a function of the number of agents N and the control parameter r. Solid lines and square points indicate the case with two areas ($M = 2$), while dashed lines and circles indicate the case with three areas ($M = 3$). Points represent the average value over 100 runs. Errorbars are smaller than the point size, and are not visible on the scale of the plot.

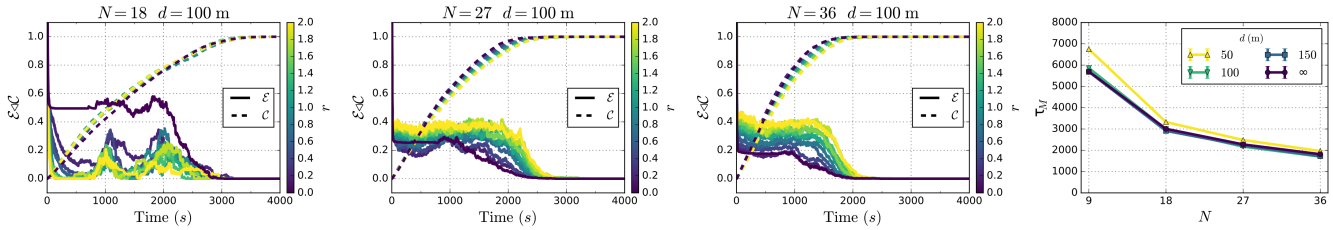

Figure 6: Results for the UAV swarm simulations. The leftmost and center panles show the, allocation errors $\mathcal{E}(t)$ for varying values of r for $N \in \{18, 27, 36\}$. The rightmost panel shows the scaling of the completion time τ_M with N for varying communication distance d. Errorbars are smaller than the point size, hence not visible on the plot.

attract agents. Together with stochastic fluctuations from the relatively small number of UAVs used, we observed a sub-optimal decision making process with respect to the area utility, as also studied in [33]. As a consequence, the deployment of UAVs to the different areas may vary across different runs (e.g., sometimes starting with A_1, others with A_2 or A_3). Hence, to study the quality of the deployment, we look at the allocation error, which is not dependent on the sequence of areas being inspected. In the top and middle panels of Figure 6, we show the case for $d = 100$ (see the supplementary material for other conditions). For $N = 18$ (leftmost panel), it is possible to note the transition from the utility-proportional deployment when $r = 0$ to the winner-take-all strategy when $r > 0$. Recruitment and cross-inhibition lead to a nearly optimal deployment as shown by the allocation error that quickly gets close to 0, although the figures are more noisy than with the multiagent simulations due to spatiality and random fluctuations. When $N = 27$ and $N = 36$ (center panels), we observe that r controls the deployment strategy in a similar way, leading to a utility-responsive deployment when $r > 0$. Communication constraints here play a double role. Mainly, they introduce additional noise and errors in the inspection, due to agents having incomplete information. However, they also reduce cross-inhibition between distant areas to attract agents, resulting in less uncommitted agents and some positive effects on the deployment.

To further understand the role of the communication constraints on the system performance, we show the average completion time τ_M over all runs and all values of r, for fixed N (see the rightmost panel in Figure 6). We note that for $d = 50$ m the completion time is higher than for the other values, especially for small N. For larger values, there is no significant deviation from the ideal case $d = \infty$. This is the result of the connectivity of the UAV network: for $d = 50$ and small N, the network is not connected and the flooding protocol does not suffice to spread information to the whole UAV swarm. In other cases, the system is beyond the percolation threshold and communication diffuses more efficiently [40].

4 CONCLUSIONS AND FUTURE WORKS

We presented a strategy for non-uniform coverage by a swarm of UAVs, and we focused on the deployment and re-deployment of UAVs to areas to be inspected. Thanks to a careful design performed at the level of the macroscopic system dynamics, it has been possible to implement an efficient decentralised algorithm resulting in a collective behaviour that can vary from utility-proportional deployment to a winner-take-all strategy, giving priority to high-utility areas and remaining responsive to changes in the area utility just by tuning a single parameter r. We have measured the allocation error of the proposed strategy with respect to an optimal allocation, and we demonstrated how it can be maintained at reasonably low values, still preserving the system responsiveness. Finally, we have shown how the strategy scales with increasing swarm size. It is worth noting that the results presented here strongly depend on the model of utility dynamics from (1), and a different picture may result from other kind of dynamics. Nevertheless, the designed strategy provides an efficient, tunable and dynamically adaptable deployment of agents to the areas to be monitored, without requiring a priori knowledge of the number of areas to be inspected, neither of their utility. More work needs to be done to generalise the proposed solution towards more general dynamics for the perceived utility of a given area.

The observed utility-responsive strategy is very relevant in view of such extensions and usage of the proposed strategy for real applications, especially for dynamic environmental conditions where utility can vary not only in response to the agents activity, but also as a result of intrinsic dynamics or external factors (e.g., epidemic spreading, logistic growth). An utility-responsive deployment can seamlessly deal with such dynamic conditions as the requirements from each area are constantly estimated and the deployment adjusted to optimise performance. Stronger ties will be pursued with the weed monitoring and mapping task, to consider utility dynamics that better fit the actual problem, and eventually deliver a field demonstration with real UAVs. Finally, a generalisation of the current approach is also desirable, to include general task allocation [18–20] and sustainable resource exploitation [38].

We have demonstrated the micro-macro link for the simplified condition of a single area to be monitored [33]. This is a fundamental result, and extensions will be attempted toward the complete system with M areas to be monitored. A deeper analysis of the macroscopic model can lead to a more informed choice of the parameterisation, advancing on the current knowledge about value-based collective decisions [32] to include dynamic environmental conditions. Additionally, adaptive tuning strategies of the key parameters should be investigated (e.g., [20]), so as to automatically attain the correct parameterisation to be used for the problem at hand.

ACKNOWLEDGMENTS

This work has been conducted within the project SAGA (Swarm Robotics for Agricultural Applications, see http://laral.istc.cnr.it/saga), an experiment founded within the ECHORD++ EU project (GA: 601116). Dario Albani and Daniele Nardi acknowledge partial support from the European project FLOURISH (GA: 644227).

REFERENCES

[1] D Albani, D Nardi, and V Trianni. 2017. Field coverage and weed mapping by UAV swarms. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 4319–4325.

[2] G Avellar, G Pereira, L Pimenta, and P Iscold. 2015. Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time. *Sensors* 15, 11 (2015), 27783–27803.

[3] S Berman, A Halasz, M A Hsieh, and V Kumar. 2009. Optimized Stochastic Policies for Task Allocation in Swarms of Robots. *IEEE Transactions on Robotics* 25, 4 (2009), 927–937.

[4] E Bonabeau, G Theraulaz, and J-L Deneubourg. 1996. Quantitative Study of the Fixed Threshold Model for the Regulation of Division of Labour in Insect Societies. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 263, 1376 (1996), 1565–1569.

[5] E Castello, T Yamamoto, F Dalla Libera, W Liu, AFT Winfield, Y Nakamura, and H Ishiguro. 2015. Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. *Swarm Intelligence* 10, 1 (2015), 1–31.

[6] EA Codling, MJ Plank, and S Benhamou. 2008. Random walk models in biology. *Journal of The Royal Society Interface* 5, 25 (2008), 813–834.

[7] N Correll. 2008. Parameter estimation and optimal control of swarm-robotic systems: A case study in distributed task allocation. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*. IEEE, 3302–3307.

[8] MB Dias, R Zlot, N Kalra, and A Stentz. 2006. Market-Based Multirobot Coordination: A Survey and Analysis. *Proc. IEEE* 94, 7 (2006), 1257–1270.

[9] C Dimidov, G Oriolo, and V Trianni. 2016. Random Walks in Swarm Robotics: An Experiment with Kilobots. In *Swarm Intelligence: 10th International Conference, ANTS 2016, Brussels, Belgium, September 7-9, 2016, Proceedings*, M Dorigo, M Birattari, X Li, M López-Ibáñez, K Ohkura, C Pinciroli, and T Stützle (Eds.). Springer International Publishing, 185–196.

[10] L Doitsidis, S Weiss, A Renzaglia, MW Achtelik, E Kosmatopoulos, R Siegwart, and D Scaramuzza. 2012. Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision. *Autonomous Robots* 33, 1-2 (2012), 173–188.

[11] K Elamvazhuthi and S Berman. 2015. Optimal control of stochastic coverage strategies for robotic swarms. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA 2015)*. IEEE, 1822–1829.

[12] J Gago, C Douthe, R E Coopman, P P Gallego, M Ribas-Carbo, J Flexas, J Escalona, and H Medrano. 2015. UAVs challenge to assess water stress for sustainable agriculture. *Agricultural Water Management* 153 (2015), 9–19.

[13] E Galceran and M Carreras. 2013. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems* 61, 12 (2013), 1258–1276.

[14] BP Gerkey and MJ Matarić. 2004. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research* 23, 9 (2004), 939–954.

[15] MA Hsieh, Á Halász, S Berman, and V Kumar. 2008. Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence* 2, 2-4 (2008), 121–141.

[16] J Keller, D Thakur, M Likhachev, J Gallier, and V Kumar. 2017. Coordinated Path Planning for Fixed-Wing UAS Conducting Persistent Surveillance Missions. *IEEE Transactions on Automation Science and Engineering* 14, 1 (2017), 17–24.

[17] PM Kietzman and PK Visscher. 2015. The anti-waggle dance: use of the stop signal as negative feedback. *Frontiers in Ecology and Evolution* 3 (2015), 14.

[18] GA Korsah, A Stentz, and MB Dias. 2013. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research* 32, 12 (2013), 1495–1512.

[19] MJB Krieger, J-B Billeter, and L Keller. 2000. Ant-like task allocation and recruitment in cooperative robots. *Nature* 406, 6799 (2000), 992–995.

[20] TH Labella, M Dorigo, and J-L Deneubourg. 2006. Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Transactions on Autonomous Adaptive Systems* 1, 1 (2006), 4–25.

[21] K Lerman, C Jones, A Galstyan, and MJ Matarić. 2016. Analysis of Dynamic Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research* 25, 3 (2016), 225–241.

[22] W Liu and AFT Winfield. 2010. Modeling and Optimization of Adaptive Foraging in Swarm Robotic Systems. *The International Journal of Robotics Research* 29, 14 (2010), 1743–1760.

[23] X Liu, Z Li, P Yang, and Y Dong. 2017. Information-centric mobile ad hoc networks and content routing: A survey. *Ad Hoc Networks* 58 (2017), 255–268.

[24] S Luke, C Cioffi-Revilla, L Panait, K Sullivan, and G Balan. 2005. MASON: A Multiagent Simulation Environment. *Simulation* 81, 7 (2005), 517–527.

[25] I Maza and A Ollero. 2007. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In *Distributed Autonomous Robotic Systems 6*. Springer Japan, Tokyo, 221–230.

[26] N Napp and E Klavins. 2011. A compositional framework for programming stochastically interacting robots. *The International Journal of Robotics Research* 30, 6 (2011), 713–729.

[27] F Nex and F Remondino. 2013. UAV for 3D mapping applications: a review. *Applied Geomatics* 6, 1 (2013), 1–15.

[28] G Pajares. 2015. Overview and Current Status of Remote Sensing Applications Based on Unmanned Aerial Vehicles (UAVs). *Photogrammetric Engineering & Remote Sensing* 81, 4 (2015), 281–330.

[29] JM Peña, J Torres-Sánchez, AI de Castro, M Kelly, and F López-Granados. 2013. Weed Mapping in Early-Season Maize Fields Using Object-Based Analysis of Unmanned Aerial Vehicle (UAV) Images. *PLoS ONE* 8, 10 (2013), e77151.

[30] A Prorok, N Correll, and A Martinoli. 2011. Multi-level spatial modeling for stochastic distributed robotic systems. *The International Journal of Robotics Research* 30, 5 (2011), 574–589.

[31] A Reina, T Bose, V Trianni, and JAR Marshall. 2018. Effects of Spatiality on Value-Sensitive Decisions Made by Robot Swarms. In *Distributed Autonomous Robotic Systems: The 13th International Symposium*, R Gross, A Kolling, S Berman, E Frazzoli, A Martinoli, F Matsuno, and M Gauci (Eds.). Springer International Publishing, Cham, 461–473.

[32] A Reina, JAR Marshall, V Trianni, and T Bose. 2017. Model of the best-of-N nest-site selection process in honeybees. *Physical Review E* 95, 5 (2017), 052411–15.

[33] A Reina, G Valentini, C Fernández-Oto, M Dorigo, and V Trianni. 2015. A Design Pattern for Decentralised Decision Making. *PLoS ONE* 10, 10 (2015), e0140950–18.

[34] SA Sadat, J Wawerla, and RT Vaughan. 2014. Recursive non-uniform coverage of unknown terrains for UAVs. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 1742–1747.

[35] SA Sadat, J Wawerla, and R Vaughan. 2015. Fractal trajectories for online non-uniform aerial coverage. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA 2015)*. IEEE, 2971–2976.

[36] TD Seeley, PK Visscher, T Schlegel, PM Hogan, NR Franks, and JAR Marshall. 2012. Stop Signals Provide Cross Inhibition in Collective Decision-Making by Honeybee Swarms. *Science* 335, 6064 (2012), 108–111.

[37] J Snape, J van den Berg, SJ Guy, and D Manocha. 2011. The Hybrid Reciprocal Velocity Obstacle. *IEEE Transactions on Robotics* 27, 4 (2011), 696–706.

[38] Z Song and RT Vaughan. 2013. Sustainable robot foraging: Adaptive fine-grained multi-robot task allocation for maximum sustainable yield of biological resources. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 3309–3316.

[39] J Torres-Sánchez, F López-Granados, N Serrano, O Arquero, and JM Peña. 2015. High-Throughput 3-D Monitoring of Agricultural-Tree Plantations with Unmanned Aerial Vehicle (UAV) Technology. *PLoS ONE* 10, 6 (2015), e0130479–20.

[40] V Trianni, D De Simone, A Reina, and A Baronchelli. 2016. Emergence of Consensus in a Multi-Robot Network: From Abstract Models to Empirical Validation. *IEEE Robotics and Automation Letters* 1, 1 (2016), 348–353.

[41] L Vig and JA Adams. 2007. Coalition Formation: From Software Agents to Robots. *Journal of Intelligent and Robotic Systems* 50, 1 (2007), 85–118.

[42] K Yu, L Qiu, J Wang, L Sun, and Z Wang. 2017. Winter wheat straw return monitoring by UAVs observations at different resolutions. *International Journal of Remote Sensing* 38, 8-10 (2017), 2260–2272.

[43] F Zhang, AL Bertozzi, K Elamvazhuthi, and S Berman. 2017. Performance Bounds on Spatial Coverage Tasks by Stochastic Robotic Swarms. *IEEE Trans. Automat. Control* (2017), 1–16.